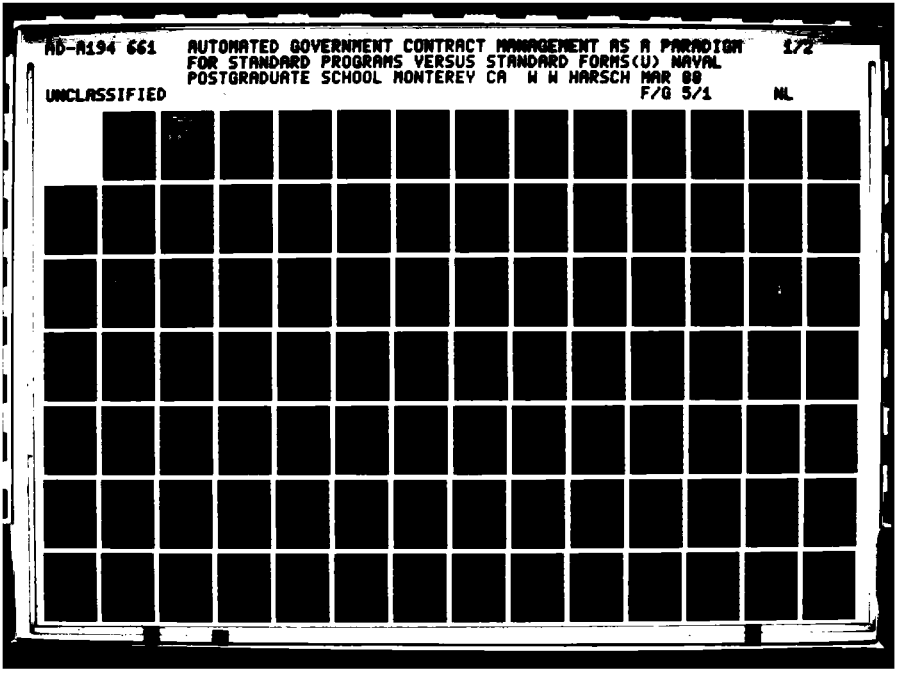
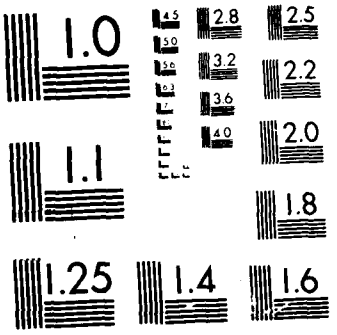

AD-A194 661 AUTOMATED GOVERNMENT CONTRACT MANAGEMENT AS A PARADIGM 1/2
FOR STANDARD PROGRAMS VERSUS STANDARD FORMS(U) NAVAL
POSTGRADUATE SCHOOL MONTEREY CA W W HARSCH MAR 88
UNCLASSIFIED F/G 5/1 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6

AD-A194 661

DTIC
ELECTE
JUN 3 0 1988
S D
C&D

NAVAL POSTGRADUATE SCHOOL
Monterey, California

DTIC FILE COPY

THESIS

AUTOMATED GOVERNMENT CONTRACT MANAGEMENT
AS A PARADIGM FOR STANDARD
PROGRAMS VS. STANDARD FORMS

by

Walter William Harsch, II

March 1988

Thesis Co-advisors: Ray Smith
Y.K. Mortagy

Approved for public release; distribution is unlimited

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

A194661

REPORT DOCUMENTATION PAGE

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) Code 54	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 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)
AUTOMATED GOVERNMENT CONTRACT MANAGEMENT AS A PARADIGM FOR STANDARD PROGRAMS VS. STANDARD FORMS

12. PERSONAL AUTHOR(S)
Harsch, Walter W. II

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Master's Thesis	FROM _____ TO _____	1988, March	110

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	
			Progress Payment; Computer Program, Theses.

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

This thesis identifies a potential weakness in the Federal Government's policy in the area of Contract Administration relating to computer prepared forms and documents. In particular, the preparation of Contract Progress Payment Requests (Standard Form 1443). It is the author's thesis that the Government, which gave us the "Standard Form," should take a leadership role in developing the "Standard Program," and that these programs be distributed to contractors free of charge in an effort to: 1. Establish and maintain program standards concerning content and documentation, and 2. Eliminate, to the maximum extent possible, mistakes in form preparation caused by math or logic errors. Keywords:

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL LCDR Ray Smith	22b. TELEPHONE (Include Area Code) (408) 646-2052 22c. OFFICE SYMBOL Code 54Sx

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

G.P.O. Government Printing Office: 1988—606-24

Approved for public release; distribution is unlimited

Automated Government Contract Management as a
Paradigm for Standard Programs vs. Standard Forms

by

Walter William Harsch, II
Lieutenant, Supply Corps, United States Navy
B.S., Metropolitan State College, 1976

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1988

Author: _____
Walter William Harsch, II

Approved by: _____
Ray Smith, Thesis Co-advisor

_____
Y.K. Mortagy, Thesis Co-advisor

_____
T.R. Sivasankaran, Second Reader

_____
David R. Whipple, Chairman
Department of Administrative Sciences

_____
James M. Fremgen,
Acting Dean of Information and Policy Sciences

ABSTRACT

This thesis identifies a potential weakness in the Federal Government's policy in the area of Contract Administration relating to computer prepared forms and documents. In particular, the preparation of Contract Progress Payment Requests (Standard Form 1443). It is the author's thesis that the Government, which gave us the "Standard Form," should take a leadership role in developing the "Standard Program," and that these programs be distributed to contractors free of charge in an effort to: 1. Establish and maintain program standards concerning content and documentation, and 2. Eliminate, to the maximum extent possible, mistakes in form preparation caused by math or logic errors.

DTIC COPY INSPECTED 6

Accesion For
NTIS CRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

TABLE OF CONTENTS

I. AN OVERVIEW ---------- 1

A. A SYNOPSIS OF THE PROBLEMS ASSOCIATED WITH USING FORMS IN GENERAL ---------- 1

B. INTUITION, INNOVATION AND LEADERSHIP -------- 3

II. BACKGROUND AND SYNOPSIS OF THE PROGRESS PAYMENT SYSTEM ---------- 7

A. DEFINITIONS AND EXPLANATIONS ---------- 7

B. CRITERIA FOR USE OF PROGRESS PAYMENTS ------- 7

C. PROGRESS PAYMENT VOLUME PER IG AUDIT -------- 8

D. SYSTEMIC "PROBLEM AREAS" ---------- 9

III. IS AUTOMATION THE ANSWER? ---------- 18

A. PRACTICAL CONSIDERATIONS ---------- 18

B. INCREMENTAL AUTOMATION ---------- 20

IV. THE "STANDARD PROGRAM" SOLUTION ---------- 31

A. SP-1443 PROGRAM FUNCTIONS AND LIMITATIONS --- 31

B. THE RELATIONAL DATABASE ---------- 33

C. STANDARDIZATION ---------- 34

D. THOUGHTS ON IMPLEMENTATION ---------- 36

E. THE FUTURE OF "STANDARD PROGRAMS" ---------- 37

V. SUMMARY ---------- 38

APPENDIX A: SP-1443 SOURCE CODE LISTING ---------- 41

APPENDIX B: SP-1443 USER'S MANUAL ---------- 88

LIST OF REFERENCES ---------- 104

INITIAL DISTRIBUTION LIST ---------- 105

I. AN OVERVIEW

This thesis will attempt to acquaint the reader with the concept of manually prepared forms in general and progress payments on Federal contracts in particular. A brief background on the purpose served by this type of contract financing will be given, including direct and indirect advantages to the government. The mechanics of how this system is administered will also be reviewed. The next major section of this thesis will advance several theories on how this system can be made to function more efficiently by using various forms of automation. Finally, a summary statement will be made, and conclusions drawn. Much of the background information for this thesis draws on the author's experience as a Contract Administrator at DCASMA Denver and as the Acting Deputy Director, Office of Telecommunications and Information Systems (OTIS) at DCASR St. Louis, Missouri.

A. A SYNOPSIS OF THE PROBLEMS ASSOCIATED WITH USING FORMS IN GENERAL

When the Government printed the first Federal Income Tax form in 1911, the intent then, as now, was to provide a means of data transfer from the taxpayer to the Government in a standardized manner. That first tax return consisted of some taxpayer identification information and only two calculations. There were no supporting schedules, or even

instruction books. The form was simple and self explanatory. In 1911 it worked very well. The Government has not changed its thinking much over the last three quarters of a century when it comes to forms. They are relatively cheap to produce, and they create a standard to which users of the form must adhere in order for the form, whatever its intended use, to be accurately processed.

The first taxpayers were asked to multiply their earnings in excess of $10,000 by 3% in order to determine the tax due if any. In sharp contrast, the Government's current Progress Payment Request Form (Standard Form 1443) contains the following instruction for only one of the over 30 data elements on the form:

> Item 20a. Of the costs reported in item 11, compute and enter only costs which are properly allocable to items that have been delivered, invoiced and accepted to the applicable date. In order of preference, these costs are to be computed on the basis of one of the following: (a) The actual unit costs of items delivered, giving proper consideration for the deferment of the starting load costs or, (b) projected unit costs (based on experienced costs plus the estimated costs to complete the contract), where the contractor maintains cost data which will clearly establish the reliability of such estimates.

Clearly, it can be seen that the complexity faced by forms users has increased dramatically. Like most major business decisions (or any other type for that matter), a periodic review would seem prudent in order to determine if the original problem is still being solved in the most expeditions manner. Unfortunately, until the current

decade, there has been no real alternative to filling out forms in order to conduct business with the Government.

As demonstrated in the instructions quoted above, the data that the Government requests may not be entirely clear to the user. In addition, there are math operations required, not just with numbers appearing on the form in question, but with cumulative values carried forward from previous forms in a series, or with figures appearing in other documents (i.e., contracts). To this already complex task, add the requirement that only whole dollars may be portrayed on the form, and that all amounts must be rounded up, and it is easy to see why so many Government forms are returned for correction and resubmission. In a June of 1987 audit report, The Naval Audit Service reported that they had audited over $41 billion in progress payments from a single Naval Plant Representative Office. The auditors concluded that $8.6 million had been over paid and that $5.9 million of that amount was due to "math errors." [Ref. 1]

B. INTUITION, INNOVATION AND LEADERSHIP

Just as individuals can take things for granted, businesses (and governments) can be lulled into a state of inactivity because the perceived priority of a given problem simply doesn't seem high enough. In 1986 Mr. Roy Rowan wrote a book entitled, The Intuitive Manager. The essence of this book is that many successful managers have acknowledged and developed an intuitive sense of what

constitutes a good course of action. One passage in particular bears repeating:

> Constantly accumulating new information and numbers, without giving the mind a chance to percolate and come to a conclusion intuitively, can delay any important decision until the time for action expires.

This quote came from the chapter titled, "Analysis Paralysis," surely something that the Government could conceivably be accused of from time to time. The relevance of this line of discussion to the forms question is this; In an era where microcomputers are being used in businesses ranging from sole proprietorships to multinational conglomerates and used for everything from word processing to balancing checking accounts, doesn't it seem a little inane to ask a contractor on a multimillion dollar contract to multiply line 5 by line 6b and type the properly rounded answer on a piece of paper?

In 1985, Peter Drucker wrote in his best seller, Innovation and Entrepreneurship:

> ...public service institutions (governments) need to build into their policies and practices the constant search for innovative opportunity. They need to view change as an opportunity rather than a threat.

This thought bears directly on the subject at hand. With progress payments in particular and all forms in general, there is a clear need for innovation in an area where intuition tells us that there must be a better way. If the Government doesn't take a leadership role soon, contractors or even third party software vendors will develop ways to

eliminate the tedium and error prone nature of hand prepared forms. In the author's opinion, there are three compelling reasons why the Government should be the innovator: 1. The Government has much to gain by streamlining the data collection process. Every time an error is detected, the form must be rejected and returned to the preparer for correction, then processed again. While this "rejection process" prevents the Government from making a questionable disbursement, it still results in processing the same request more than once. The ideal situation is to have the form correct in the first place. 2. By virtue of its size and sovereignty, the Government can establish and enforce the standards needed to make such a system compatible with existing hardware and software. 3. The Government makes the rules, and prints the current manual forms. Who would be in a better natural position to maintain and revise software to keep pace with current legislative and regulatory changes?

In February of 1988, LCDR John Grassi, SC, USN, speaking to a roundtable of Internal Auditors, discussed the concept of leadership in terms of integrity, competence and vision. It is precisely these qualities that mandate the need for creative change in the way the government conducts its day to day business with the private sector. Vision in particular seems to be a valuable commodity when pondering any departure from the status quo. As defined by LCDR

Grassi, vision is sensing destiny through the details that concern us. In 1911, there were no viable alternatives to the manually prepared form; in 1988 there are. These alternatives should be defined, examined and acted upon, "...before the time for action expires."

II. BACKGROUND AND SYNOPSIS OF THE PROGRESS PAYMENT SYSTEM

A. DEFINITIONS AND EXPLANATIONS

The Federal Government frequently contracts for goods or services where the deliverables are not tendered to the buyer until near the end of the contractual period. In the interim, the contractor may well be accumulating significant allowable and allocable costs that are directly related to the contract at hand but, without some form of Government contract financing, will not ordinarily be reimbursed until delivery. By providing contract financing, the Government can exert more influence in the marketplace by giving certain contractors a chance to compete for contracts that they would otherwise not be able to finance. This helps the Government to stimulate competition, support Small Business, broaden the industrial base and promote a higher level of performance. One form of Government financing is the use of progress payments. Authorized in FAR 32.5, a contracting officer may, under certain circumstances, include progress payments in certain fixed price type contracts.

B. CRITERIA FOR USE OF PROGRESS PAYMENTS

FAR 32.502 offers guidance to the contracting officer in deciding when progress payments may be authorized.

> ...the contracting officer may provide for customary progress payments if the contractor 1) will not be able to

bill for the first delivery...for a substantial time after the work must begin (normally 4 months or more for small business concerns; 6 months or more for others), and 2) will make expenditures for contract performance during the predelivery period that have a significant impact on the contractor's working capital....To reduce undue administrative effort and expense, the contracting officer generally should not provide for progress payments on contracts of less than $1,000,000 unless (1) The contractor is a s small business concern and the contract will involve approximately $100,000 or more; or (2) The contractor will perform a group of small contracts at the same time and the total impact on working capital is equivalent to a single contract of $1,000,000 or more.

C. PROGRESS PAYMENT VOLUME PER IG AUDIT

To give the reader an idea of the magnitude of the dollars that flow through this system, a 1984 Inspector General Report [Ref. 2] estimated that in 1983, over 56 billion dollars were disbursed to military contractors in the form of progress payments. The percentage of incurred costs that are eligible for regular progress payments has varied over the years, roughly coinciding with major changes in the cost of money in the private sector. The current rates are 75% for large businesses and 80% for small businesses. It is not within the scope of this thesis to discuss these reimbursement rates, contractor eligibility requirements or even whether or not there should even be progress payments. The primary focus will be on systemic efficiency and possible automated system improvements.

Ostensibly, progress payments are simply a means to finance long term, high dollar value contracts. The government in effect plays the roll of working capital

provider. The government's investment is protected by the eligibility requirements placed on contractors, and by the passage of title to work in process to the government.

Since one of the eligibility criteria for a contractor to receive progress payments is the certification that an adequate accounting system is in place [Ref. 3] to accurately accumulate costs, the attendant Defense Contract Audit Agency (DCAA) accounting system review gives the government virtually unrestricted access to a contractor's books. This may be a key factor in cases where a contractor is too small, or does not do enough government business to warrant audit procedures. By applying for progress payments, a contractor is effectively opening the door to DCAA. After the initial review, the contracting officer may elect to audit any or all subsequent progress payment requests. In some cases, the contracting officer may elect to have DCAA conduct a post-payment audit on an individual request (or prior to payment if there is fraud suspected).

D. SYSTEMIC "PROBLEM AREAS"

One of the costs to the government in running the progress payment system, is the cost of administration. There are one time (per contract) costs to set the system in motion. These functions do not readily lend themselves to automation. Each time a contractor requests payment however (usually monthly, more often in special cases), a fairly mechanical, repetitive and well defined sequence of events

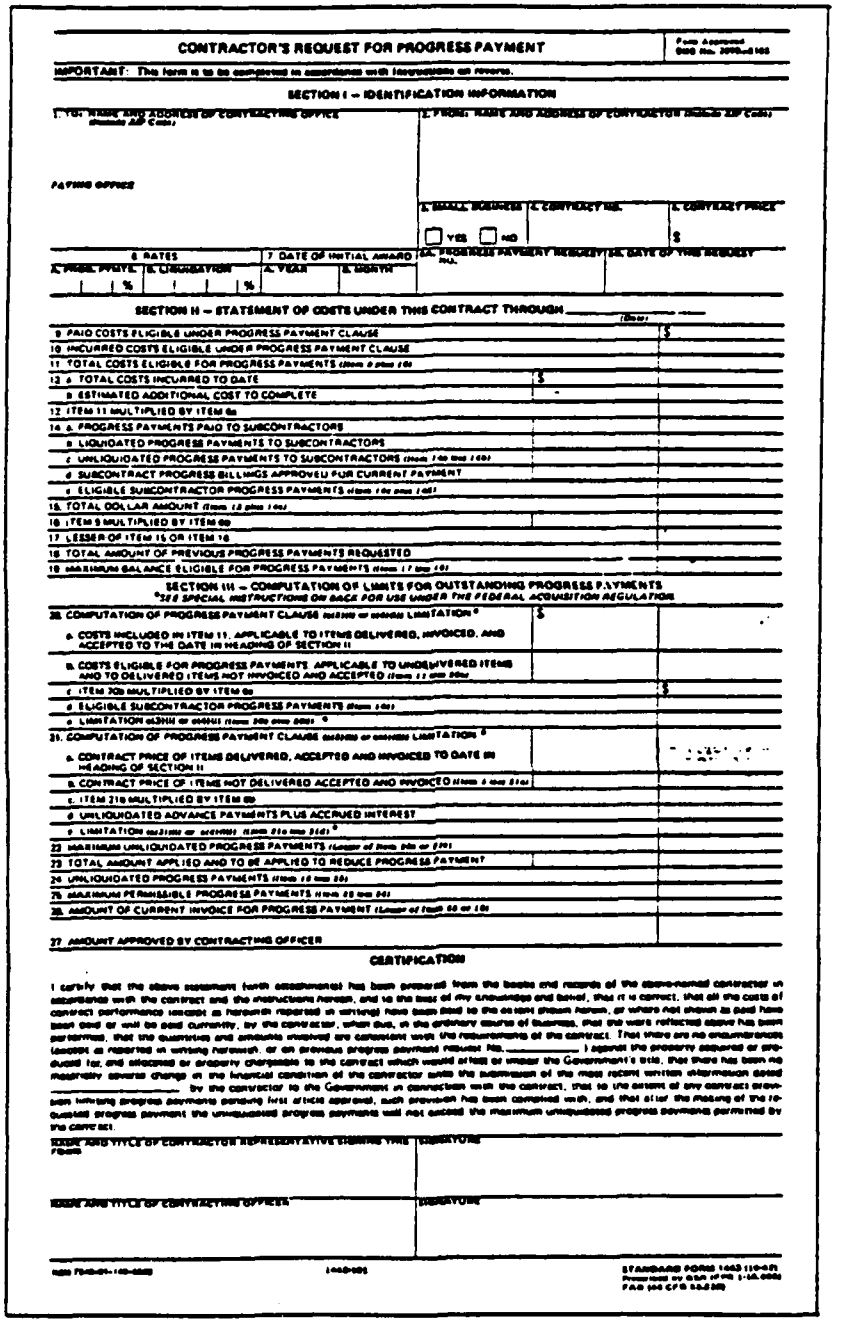
must take place. This process centers around the preparation, submission and payment of a progress payment request, Standard Form 1443, and does offer automation potential.

1. Complex Forms

The SF 1443 is a single legal size sheet of paper that resembles a tax return (see Fig. 1). The front of the form consists of three sections. The first is identification information (administration office, paying office, and contractor and contract data. The second section contains the financial data. The third section is a certification signed by the contractor that the amounts are both allowable and allocable to the contract and that all restrictions and requirements have been observed. There are signature blocks for the contractor and the contracting officer approving the payment. The back of the form has fairly detailed instructions.

a. Primary Data

Sections 2 and 3 of the form contain the numerical "meat" of the request. There are 32 blanks that the contractor must enter with appropriate dollar values. It is important to make a distinction between "primary" data and "secondary" information. Primary data are the amounts on the request that constitute what the contractor is asking for. There are a maximum of eight lines on the SF 1443 that require actual contractor input, and nearly half of those apply only if the contractor utilizes subcontractors.



CONTRACTOR'S REQUEST FOR PROGRESS PAYMENT

IMPORTANT: This form is to be completed in accordance with instructions on reverse.

SECTION I – IDENTIFICATION INFORMATION

1. TO: NAME AND ADDRESS OF CONTRACTING OFFICE (Include ZIP Code)

PAYING OFFICE

2. FROM: NAME AND ADDRESS OF CONTRACTOR (Include ZIP Code)

3. SMALL BUSINESS ☐ YES ☐ NO | 4. CONTRACT NO. | 5. CONTRACT PRICE $

6. RATES		7. DATE OF INITIAL AWARD		8A. PROGRESS PAYMENT REQUEST NO.	8B. DATE OF THIS REQUEST
A. PUBLIC VOUCHER	B. LIQUIDATION	A. YEAR	B. MONTH		
%	%				

SECTION II – STATEMENT OF COSTS UNDER THIS CONTRACT THROUGH _____ (Date)

9. PAID COSTS ELIGIBLE UNDER PROGRESS PAYMENT CLAUSE		$
10. INCURRED COSTS ELIGIBLE UNDER PROGRESS PAYMENT CLAUSE		
11. TOTAL COSTS ELIGIBLE FOR PROGRESS PAYMENTS (Sum of items 9 and 10)		
12. a. TOTAL COSTS INCURRED TO DATE	$	
b. ESTIMATED ADDITIONAL COST TO COMPLETE		
13. ITEM 11 MULTIPLIED BY ITEM 6a		
14. a. PROGRESS PAYMENTS PAID TO SUBCONTRACTORS		
b. LIQUIDATED PROGRESS PAYMENTS TO SUBCONTRACTORS		
c. UNLIQUIDATED PROGRESS PAYMENTS TO SUBCONTRACTORS (Item 14a less 14b)		
d. SUBCONTRACT PROGRESS BILLINGS APPROVED FOR CURRENT PAYMENT		
e. ELIGIBLE SUBCONTRACTOR PROGRESS PAYMENTS (Item 14c or 14d)		
15. TOTAL DOLLAR AMOUNT (Items 13 plus 14e)		
16. ITEM 5 MULTIPLIED BY ITEM 6b		
17. LESSER OF ITEM 15 OR ITEM 16		
18. TOTAL AMOUNT OF PREVIOUS PROGRESS PAYMENTS REQUESTED		
19. MAXIMUM BALANCE ELIGIBLE FOR PROGRESS PAYMENTS (Item 17 less 18)		

SECTION III – COMPUTATION OF LIMITS FOR OUTSTANDING PROGRESS PAYMENTS
*SEE SPECIAL INSTRUCTIONS ON BACK FOR USE UNDER THE FEDERAL ACQUISITION REGULATION

20. COMPUTATION OF PROGRESS PAYMENT CLAUSE (lesser or applicable LIMITATION)*	$	
a. COSTS INCLUDED IN ITEM 11, APPLICABLE TO ITEMS DELIVERED, INVOICED, AND ACCEPTED TO THE DATE IN HEADING OF SECTION II		
b. COSTS ELIGIBLE FOR PROGRESS PAYMENTS APPLICABLE TO UNDELIVERED ITEMS AND TO DELIVERED ITEMS NOT INVOICED AND ACCEPTED (Items 11 less 20a)		
c. ITEM 20b MULTIPLIED BY ITEM 6b		$
d. ELIGIBLE SUBCONTRACTOR PROGRESS PAYMENTS (Item 14e)		
e. LIMITATION (lesser of 20c or 20d)(Items 20c less 20d)*		
21. COMPUTATION OF PROGRESS PAYMENT CLAUSE (lesser or applicable LIMITATION)*		
a. CONTRACT PRICE OF ITEMS DELIVERED, ACCEPTED AND INVOICED TO DATE IN HEADING OF SECTION II		
b. CONTRACT PRICE OF ITEMS NOT DELIVERED ACCEPTED AND INVOICED (Items 5 less 21a)		
c. ITEM 21b MULTIPLIED BY ITEM 6b		
d. UNLIQUIDATED ADVANCE PAYMENTS PLUS ACCRUED INTEREST		
e. LIMITATION (sum of 21c and 21d) (Items 21c less 21d)*		
22. MAXIMUM UNLIQUIDATED PROGRESS PAYMENTS (Lesser of Items 20e or 21e)		
23. TOTAL AMOUNT APPLIED AND TO BE APPLIED TO REDUCE PROGRESS PAYMENT		
24. UNLIQUIDATED PROGRESS PAYMENTS (Item 18 less 23)		
25. MAXIMUM PERMISSIBLE PROGRESS PAYMENTS (Item 22 less 24)		
26. AMOUNT OF CURRENT INVOICE FOR PROGRESS PAYMENT (Lesser of Item 19 or 25)		
27. AMOUNT APPROVED BY CONTRACTING OFFICER		

CERTIFICATION

I certify that the above statement (with attachments) has been prepared from the books and records of the above-named contractor in accordance with the contract and the instructions hereon, and to the best of my knowledge and belief, that it is correct, that all the costs of contract performance (except as herein reported or in writing) have been paid to the extent shown herein, or where not shown as paid have been paid or will be paid currently, by the contractor, when due, in the ordinary course of business, that the work reflected above has been performed, that the quantities and amounts involved are consistent with the requirements of the contract. That there are no encumbrances (except as reported in writing herewith, or on previous progress payment requests No. _____) against the property acquired or produced for, and allocated or properly chargeable to the contract which would affect or impair the Government's title, that there has been no materially adverse change in the financial condition of the contractor since the submission of the most recent written information dated _____ by the contractor to the Government in connection with the contract, that to the extent of any contract provision limiting progress payments pending final article approval, such provision has been complied with, and that after the making of the requested progress payment the unliquidated progress payments will not exceed the maximum unliquidated progress payments permitted by the contract.

NAME AND TITLE OF CONTRACTOR REPRESENTATIVE SIGNING THIS REQUEST | SIGNATURE

NAME AND TITLE OF CONTRACTING OFFICER | SIGNATURE

1443-005

STANDARD FORM 1443 (10-87)
Prescribed by GSA FAR (48 CFR) 53.232-16

Figure 1. Standard Form SP-1443

b. Secondary Data

The other 24 lines are secondary data; mathematical manipulations of primary data and data from previous progress payment requests on the same contract if any. It is this secondary data that offers the greatest potential for automation. The regulations governing their computation are fairly straight forward and easy to write into a program. This is also the most tedious function in the preparation of the form. To insure complete accuracy, cumulative figures must be calculated from the beginning of the contract to the present. Since the contract can be modified or amended, prices and quantities for deliverables may vary, making it even more difficult to reconcile. Here is a prime example of the mind set that created the form in the first place; the Government's method for controlling these payments is based on the cumulative "to date" figures appearing on the form. The contractor on the other hand, prepares the request from his accounting records that are logically divided by months (in most cases). The contractor must extract the current monthly information that will constitute the basis for his request then add those numbers to the cumulative prior period figures to arrive at the numbers that will appear on the form. We see here, the same problem viewed from two different vantage points. The problem remains the same (assumption: it is factual) but the logical "views" are different. This is precisely the

type of situation that the modern relational database management software is designed to resolve.

c. Contractor Certification

The Section I information will remain the same for each request except for date and progress payment sequence number. The Section II and III data will be computed and verified with each request. The certification statement and signature blocks however, present an automation problem, but not an unsolvable one.

d. Authorization Signatures

The contractor's signature on the progress payment request attests to the document's compliance with the contract and its legitimacy as mentioned on the previous page. The contracting officer has a block assigned for the amount of money he or she is willing to approve for the request (which may or may not be the amount that the contractor is requesting), and a signature block. The contracting officer is a "warranted" agent of the United States. While serving in this fiduciary capacity, he is legally liable for the consequences of his actions. The contracting officer is tasked with reviewing the form for completeness and accuracy.

2. Math Accuracy and Validation

In addition to the "allowability and allocability" issue, the form must be mathematically correct before it can be honored. This seemingly simple task is complicated in

two ways. First, many of the 32 amounts are tied to the previous progress payment request. Some figures are cumulative and are carried forward from one payment to the next in sequence. Second, the price and or quantity of the individual line items (and hence the contract), may fluctuate via contract modification. It is possible for a contractor to have signed a bilateral contract modification prior to submitting a progress payment request, and still have that request rejected because the contract mod price changes were not "booked" prior to review of the progress payment request. As a practical matter, some contractors include copies of recent modifications if a timing problem is thought to exist. Simple procurements are not generally a problem in this area, but 70 to 100 modifications are not at all uncommon on larger procurements.

To further complicate the math accuracy problem, progress payment requests use an unusual rounding technique. All block entries are required to be in whole dollars. They are always rounded up, i.e., the amount $12.01 will appear as $13. This seems simple, but it is not "conventional," and indeed federal state and local tax returns by convention round down at the $.49 level. Many progress payment requests are sent back to contractors for a $1 error.

The exact routing of progress payment requests may vary slightly from one activity to the next, but in essence, the form is prepared and signed by the contractor, submitted

to the contracting officer for review and approval, then submitted to the paying office for validation and payment. That last step can be a lengthy one. DCASR's for example have data entry clerks type the pertinent data elements into a computer line for line from the approved request. (Author's Note: In 1981, the Defense Logistics Agency implemented the Automated Progress Payment Program in all of their DCASR's. This system is a computer program installed and run on the individual DCASR mainframe computers to determine the "validity" of contractor progress payment requests prior to payment. Other DoD activities may or may not have such an automated system in place.) The clerks are not tasked with finding and correcting errors or omissions, even if they are obvious. Their job is to simply transcribe the data into a computer. Even if a clerk wanted to correct one of these errors, he or she would lack legal capacity to do so.

3. Consistency

The computer, linked to the contract payment data base, performs a series of checks. This process is called "validation." The computer compares the current request with all previous SF 1443's to insure consistency and mathematical accuracy. Requests are serialized to detect skipped or duplicate payments. If a request fails validation it returned to the contractor and a notification is sent to the contracting officer. The process then starts

again. DCASR St. Louis experienced a rejection rate of about one in four progress payment requests on average, with some companies having fewer approvals than rejections.

4. Visibility Within the System

From the time the progress payment request leaves the contractor to the time it is first entered into the computer, the document has a fairly low visibility within the government "system." If a contractor calls the contracting office for status, the answer will usually require several subsequent phone calls to various departments. This creates additional administrative overhead for the government, and for contractors.

5. Timeliness of Payments

By their very nature, progress payments are rarely for less than a thousand dollars. Usually they are in the tens to hundreds of thousands range, sometimes higher. Most agencies exempt progress payments from the cash management program. Withholding payment for 30 days would, in effect, dilute the effectiveness of progress payments in the first place. The fact remains that processing the requests for payment requires considerable paper handling and review before disbursement can be authorized. The promptness of these payments can be a critical issue with contractors. One Colorado Springs aerospace industry opened a bank account in St. Louis just to save the one or two day mail delay for checks. It is the author's thesis that

streamlining the progress payment system will have a dual benefit. The first would be to reduce the rejection rate of requests, reducing reprocessing costs and eliminating unnecessary delays in disbursing/receiving payment. The second would be to better facilitate (in the long run) the overall objective of Defense procurement; to insure an adequate flow of quality goods and services to the Department of Defense at a reasonable cost to the taxpayer. A contractor is more likely to bid on a contract when the procedure for payment is well established and predictable.

III. IS AUTOMATION THE ANSWER?

A. PRACTICAL CONSIDERATIONS

1. Availability of ADP Capacity

Several practical considerations impact any decision to automate any manual system. The first is ADP capacity. In many cases there may be excess computer capacity available on site, with a clear path to procuring additional hardware or software as required. In other cases there may be limitations that will be difficult to overcome. The key would seem to be reasonableness. Sometimes the best solution to an information transfer problem is still a typewriter and a piece of paper. Just because an automated solution is possible does not make it economically advisable. This concept applies equally to government and industry.

2. Resistance to Change

In many offices, particularly in employees that received their educations prior to the "computer age," there is sometimes a resistance to change. When the change involves computers and other electronic gadgets, the resistance usually gets even stronger. Training offers a solution to this impediment. Lecture sessions and handouts are not enough. People must be made to realize that they are still the key to problem solving, and personal one on

one practical training may be needed to convince some people. Sometimes a simple demonstration is all it takes to convince a nonbeliever. The author feels that this is a very important issue. As more and more mundane and repetitive tasks are relegated to machines, the quality of the work done by people becomes proportionally more critical.

3. Legal Considerations

Finally we come to legal restrictions. A properly signed and dated progress payment request carries certain legal status. A senior official of the company personally attests to the accuracy of the document and its compliance with current regulations. A duly authorized agent of the government places his or her name and professional reputation on the line when they sign. It is important however to note that the progress payment request does not meet the criteria for a contract. There is not a promise for an act or the forbearance of an act. Not even a promise for a promise. A progress payment request is not a negotiable instrument, it is simply a request for payment. The certification section of the form is simply a restatement of the obligations of the contractor. If it were stricken from the form altogether, a case could be made that it would not reduce the liability of the company at all if fraud were detected. In the case of fraud, the signed

certification does provide an additional remedy, and is generally easier to prove than the "intent" to defraud.

4. Availability of Solutions

The technology exists today to totally automate the progress payment process. As already discussed, that does not make total automation a near term, or for that matter even a long term goal. The following paragraphs will explore some of the author's conceptual solutions and discuss the "proof of concept" program (SP-1443) in Appendix A.

B. INCREMENTAL AUTOMATION

As discussed earlier, the information section of the SF 1443 remains essentially unchanged from one payment request to the next. The 32 amounts found in Sections 2 and 3 could be reduced to no more than eight if a computer were used to derive secondary data. A computer software product that knows about the round up rule as well as the other "validation" steps used by government paying offices would never make a simple math error. If properly written, the program could alert the contractor to upcoming thresholds and limitations before they become a critical issue. Modern software does not have to ask the same question over and over if it already knows the answer. For example; if during initial set up, a contractor answers the "Are you a Small Business?" with "Yes," then the entry on line #9 (Paid costs eligible...) will always be $0. The software knows that

Small Businesses are reimbursed for incurred costs and that line #9 is not used. Using a microcomputer based progress payment system would have several advantages to the contractor; a clear, concise record of a progress payment transactions for a given contract, elimination of logical as well as mathematical errors, and reduction in preparation time just to name a few.

1. Computer Generated SF-1443

There are several ways to implement this concept. One Boulder Colorado firm used Lotus 123 (tm) and created a "spreadsheet" to generate secondary data and maintain continuity. Eventually the firm was successful in generating a report on a dot matrix printer that resembled the original enough to be accepted by the paying office. Not every company will develope independent software solutions like this, but some will. Once the decision has been made by a contractor to automate some portion of the contract management function, the question then becomes one of degree. Should the program stand alone, or should it be integrated with other accounting and management software? The advantage to the government of independant software development like this is that the contractor/developer remains responsible for program defects or flaws. In a Government supplied program, at least some of that responsibility would shift and the contractor may be able to make a case that errors or omissions on an electronically

prepared form may be the fault of the software. Herein, however, lies one of the dangers of letting a contractor or third party vendor create software. The resulting form may look correct at first inspection, it may even "validate" in the DLA APPP system. What the program does in unforseen (and unplanned for by the programmer) situations cannot be predicted by the government unless it conducts exhaustive tests on every piece of software submitted to it. For example, when the estimate to complete a given contract exceeds the difference between the contract price and the amount already paid, the contract is said to be in a "loss position." The net effect of this is that a contractor's reimbursement rate is reduced by an amount sufficient to insure that the government will not pay out in progress payments a sum greater than the price of the contract. How should the program react to a loss situation? Should it inform the contractor that the situation exists? Should it ignore the possibility by simply "backing into" the estimate to complete figure rather than asking the contractor to insert it? The point is, that the Government is the entity that should make these decisions in order to maintain accuracy and consistency. Again, the need for exhaustive testing is stressed in order to avoid contractor claims that Government provided software may result in an error. Figure 2 is a SF-1443 created by the SP-1443 program.

IMPORTANT: This form is to be completed in accordance with instructions on reverse.

SECTION I - IDENTIFICATION INFORMATION

1. TO: NAME AND ADDRESS OF CONTRACTING OFFICE	2. FROM: NAME AND ADDRESS OF CONTRACTOR (Include ZIP Code)
DCASMA Denver 750 W. Hampden Ave. Englewood CO 80220-0000	The ABC Sunshine Co. 2233 Main St. Boulder CO 80330-0001
PAYING OFFICE: DCASR St. Louis 1139 Washington St. St. Louis MO 69361-0000	

3. SMALL BUSINESS	4. CONTRACT NUMBER	5. CONTRACT PRICE
[ ]YES [X]NO	N00000-87-C-0100	$ 1,000,000

6. RATES		7. DATE OF INITIAL AWARD		8A. PROGRESS PAYMENT REQUEST NO.	8B. DATE OF THIS REQUEST
A. PROG. PYMTS.	B. LIQUIDATION	A. YEAR	B. MONTH		
75.0 %	75.0 %	1987	11	2	03/29/88

SECTION II - STATEMENT OF COSTS UNDER THIS CONTRACT THROUGH 01/15/88

9. PAID COSTS ELIGIBLE UNDER PROGRESS PAYMENT CLAUSE		$ 200,000
10. INCURRED COSTS ELIGIBLE UNDER PROGRESS PAYMENT CLAUSE		0
11. TOTAL COSTS ELIGIBLE FOR PROGRESS PAYMENTS (Item 9 plus 10)		200,000
12a. TOTAL COSTS INCURRED TO DATE	$ 200,000	
b. ESTIMATED ADDITIONAL COST TO COMPLETE	800,000	
13. ITEM 11 MULTIPLIED BY ITEM 6a.		150,000
14a. PROGRESS PAYMENTS PAID TO SUBCONTRACTORS	0	
b. LIQUIDATED PROGRESS PAYMENTS TO SUBCONTRACTORS	0	
c. UNLIQUIDATED PROGRESS PAYMENTS TO SUBCONTRACTORS (Item 14a less 14b)	0	
d. SUBCONTRACT PROGRESS BILLINGS APPROVED FOR CURRENT PAYMENT	0	
e. ELIGIBLE SUBCONTRACTOR PROGRESS PAYMENTS (Item 14c plus 14e)		0
15. TOTAL DOLLAR AMOUNT (Item 13 plus 14e)		150,000
16. ITEM 5 MULTIPLIED BY ITEM 6b	750,000	
17. LESSER OF ITEM 15 OR ITEM 16		150,000
18. TOTAL AMOUNT OF PREVIOUS PROGRESS PAYMENTS REQUESTED		75,000
19. MAXIMUM BALANCE ELIGIBLE FOR PROGRESS PAYMENTS (Item 17 less 18)		75,000

SECTION III - COMPUTATION OF LIMITS FOR OUTSTANDING PROGRESS PAYMENTS
* SEE SPECIAL INSTRUCTIONS ON BACK FOR USE UNDER THE FEDERAL ACQUISITION REGULATION.

20. COMPUTATION OF PROGRESS PAYMENT CLAUSE (a(3)(i) or a(4)(i)) LIMITATION *		
a. COSTS INCLUDED IN ITEM 11, APPLICABLE TO DELIVERED, INVOICED, AND ACCEPTED TO THE DATE IN HEADING OF SECTION II.	$ 0	
b. COSTS ELIGIBLE FOR PROGRESS PAYMENTS, APPLICABLE TO UNDELIVERED ITEMS AND TO DELIVERED ITEMS NOT INVOICED AND ACCEPTED (Item 11 less 20a)	200,000	
c. ITEM 20b MULTIPLIED BY 6a		$ 150,000
d. ELIGIBLE SUBCONTRACTORS PROGRESS PAYMENTS (Item 14e)		0
e. LIMITATION a(3)(i) or a(4)(i) (Item 20c plus 20d) *		150,000
21. COMPUTATION OF PROGRESS PAYMENT CLAUSE (a(3)(ii) or a(4)(ii)) LIMITATION *		
a. CONTRACT PRICE OF ITEMS DELIVERED, ACCEPTED AND INVOICED TO DATE IN HEADING OF SECTION II	0	
b. CONTRACT PRICE OF ITEMS NOT DELIVERED, ACCEPTED AND INVOICED (Item 5 less 21a)	1,000,000	
c. ITEM 21b MULTIPLIED BY ITEM 6b		750,000
d. UNLIQUIDATED ADVANCE PAYMENTS PLUS ACCRUED INTEREST		0
e. LIMITATION (a(3)(ii) or a(4)(ii)) (Item 21c less 21d) *		750,000
22. MAXIMUM UNLIQUIDATED PROGRESS PAYMENTS (Lesser of Item 20e or 21e)		150,000
23. TOTAL AMOUNT APPLIED AND TO BE APPLIED TO REDUCE PROGRESS PAYMENT	0	
24. UNLIQUIDATED PROGRESS PAYMENTS (Item 18 less 23)		75,000
25. MAXIMUM PERMISSIBLE PROGRESS PAYMENTS (Item 22 less 24)		75,000
26. AMOUNT OF CURRENT INVOICE FOR PROGRESS PAYMENT (Lesser of Item 25 or 19)		75,000
27. AMOUNT APPROVED BY CONTRACTING OFFICER		

CERTIFICATION

I certify that the above statement (with attachments) has been prepared from the books and records of the above-named contractor in accordance with the contract and the instructions hereon, and to the best of my knowledge and belief, that it is correct, that all the costs of the contract performance (except as herewith reported in writing) have been paid to the extent shown herein, or where not shown as paid have been paid or will be paid currently, by the contractor, when due, in the ordinary course of business, that the work reflected above has been performed, that the quantities and amounts involved are consistent with the requirements of the contract. That there are no encumbrances (except as reported in writing herewith, or on previous progress payment request No.) against the property acquired or produced for, and allocated or properly chargeable to the contract which would affect or impair the Government's title, that there has been no materially adverse change in the financial condition of the contractor since the submission of the most recent written information dated by the contractor to the Government in connection with the contract, that to the extent of any contract provision limiting progress payments pending first article approval, such has been complied with, and that after the making of the requested progress payments the unliquidated progress payments will not exceed the maximum unliquidated progress payments by the contract.

NAME AND TITLE OF CONTRACTOR REPRESENTATIVE SIGNING THIS FORM	SIGNATURE
	DRAFT COPY ONLY, NOT FOR SUBMISSION !!!
NAME AND TITLE OF CONTRACTING OFFICER	SIGNATURE

Figure 2. Computer-generated Standard Form 1443

2. Contracting Officer Automated Review

As discussed earlier, DLA implemented an automated progress payment review program in 1981 to run on mainframe computers. The intent of this effort was to reduce the workload on Contracting Officers by relieving them of the tedious mathematical and data retrieval aspects of manual review. The inherent flaw with the APPP system, is that it did not "alert" on an erroneous request until the form was forwarded to the paying office and keyed into the computer by a clerk. If a Contracting Officer really wanted to be confident that the request was not in error, a complete manual calculation and review would be necessary. Consider the value that a real time, accurate and complete payment request review would have. Not only would the error be caught earlier in the cycle, but the person detecting the error (the contracting officer) would have the knowledge and authority to make a correction if it were necessary. If a SF-1443 generating program makes a logical "first step" in automating this process, then would not using a variation of the program to allow the Government to review requests make a logical "second step"?

3. Electronic Transmission of Requests

Because of the inherent high dollar value of progress payment requests, many contractors sent these requests by an express mail service to insure "overnight" delivery. This service can cost over $14 for a single form!

Since nearly all Government contracting facilities are equipped with the capacity to receive and transmit commercial messages (Telex, TWX, Western Union, etc., for the purpose of accepting telegraphic bids) does it not make intuitive sense to transmit progress payment requests via that medium? A number of vendors, including Western Union, offer at a nominal charge (about $.45 for a document the length of a progress payment request) the ability to transmit a prerecorded message from a computer, via telephone modem, directly into the commercial message network. A logical "third step" perhaps? Here again the subject of signatures and certifications can be raised. The contractor may still be able to "certify" the request by using a predetermined code or password. The legality of such practices is beyond the scope of this thesis, but it is notable that electronically synthesized signatures have been used in other systems (checks, contracts etc.). The passage of time and the courts will have to rule on the validity of such methods.

4. Automatic Review and Payment

Since nearly all DLA administered contracts have been validated and paid by computer since 1981, and the ability exists now to create, review and transmit the requests electronically, it seems a short step indeed to eliminate the manual data entry process at the paying office as well and forward the "electronic" request directly to the

paying computer. There should be prudent safeguards of course. General "dial up" capability into a computer is usually thought of as a weakening of security. Perhaps an intermediate computer functioning much like an electronic bulletin board could receive the requests, log them in and do an initial screen to see that they are indeed valid requests. Once received and the caller disconnected, the information could be uploaded into the main computer for processing. This process would save additional time and avoid the possibility of a transposed number or other error on manual entry. This would constitute the "fourth step."

5. Variations on the Theme

The previous four subheads have taken a somewhat idealized look at the potential for incremental automation using a microcomputer and some form of commercial telecommunication. The "hard copy" form that would be generated would be similar to a SF-1443, but if the telegraphic concept were adopted, it would probably be abbreviated and would, of course, lack the signatures of the contractor's representative and the Administrative Contracting Officer. As mentioned previously, there are ways to accomplish the same objective electronically. The contractor's certification and the authorizing signatures are the only aspects of the current SF-1443 that could not be retained in the methodology proposed above.

Setting aside these two drawbacks for a moment, what possible benefits could accrue to the government from such a system? For one thing, once something has been recorded electronically, there is no need to retype it. In theory, an incoming message of this type could be recorded on a disk just as easily as printing it. As an alternative to purely electronic processing, a diskette could be prepared by the contractor and forwarded to the Contracting Officer together with a hard copy print out bearing the certification and signature of the contractor. If the contracting officer had a copy of the same (or similar) software, much of the review process could be conducted automatically. The contracting officer could then enter the amount approved for payment to line #27, sign the hard copy and forward both the hard copy and the diskette to the paying office. The hard copy could be held for backup purposes, and the diskette used to enter the request into the computer instead of keying it in. Although not as "streamlined" as the previous methodology, this system could still result in substantial administration time savings, but perhaps the biggest savings would come from the lowering of the rejection rate because the request would be inherently more accurate in the first place.

For those who find it difficult do disburse funds solely on the basis of unseen magnetic imagery there is at least one more basic approach that might offer hope. This would be to retain the hard copy, computer generated SF-1443

complete with human signatures, and use optical scanners for the review and validation phases. Such technology is available today. The text appearing in Figures 3 and 4 was not "typed" in by the author, it was actually "scanned" into a computer using equipment in one of the computer labs at the Naval Postgraduate School. This solves the signature problem, but still requires paper handling by humans.

Implementation of DODD 5500.7

Developed by MN 4371 Policy Class

#1 1. Top management must set the example and insist on keeping issue alive throughout organization (14)

#3 2. Emphasize enforcement (14)

3. Centrally managed policy--unified standards (12)

4. Familiarize employees with real temptations in the workplace; be specific (12)

Figure 3. Scan Document #1

A Portion of text generated on a dot matrix printer and scanned into a word processing text file. The (12) was relocated to the next line because of margin settings for this thesis. A portion of a DoD Directive concerned Standards of Conduct. This document was a photocopy of the actual report and was scanned as Figure 4. The periodic # symbol indicates an unreadable character.

SUBJECT: Standards of Conduct

References: (a) DoD Directive 5500.7, subject as above, January 15, 1977 (hereby canceled) (b) DoD Directive 7700. 15, ''Reporting Procedures on De#en#e Related Employment,'' October 30, 1970 (hereby canceled)
(c) Public I#w 95-521, "Ethics in Government Act of 1978," October 26, 1978, as amended
(d) Title 5, Code of Federal Regulations, Parts. 734 and 735, Office of Personnel Management Regulations
(e)through (jj) , see enclosure 1

A. #####;UANCE AND #U##OSE

1. This Directive reissues reference (a) after reference (a) was consolidated with reference (b), and implements references (c) through (f).

2. This Directive prescribes standards of conduct required of all DoD personnel, regardless of assignment. It establishes criteria and procedures for reports required of certain former and retired military officers and former DoD civilian officers and employees who are presently employed by defense contractors,
and former officers and employees of defense contractors present-
ly employed by the Department of Defense.

Figure 4. Scan Document #2

Discussions so far have centered around a DCASR/DCASMA environment because of the author's background. The concepts however should be adaptable to Plant Representative Offices and major buying commands as well. The use of commercial telecommunications is not essential at all, but is currently available. The Defense Data Network (DDN) has a mature and sophisticated electronic mail handling capability, and would be well suited to this

application. Indeed it will probably be required for any long range telecommunications within a few years (DLA is currently operating on a waiver from DCA and leases commercial trunk lines to form the DLA net). Facsimile transmission is currently in vogue for many businesses and is used extensively by the government. In point of fact however, data can be sent with greater accuracy and at much higher speeds (less cost) with DDN or DDN like systems.

IV. THE "STANDARD PROGRAM" SOLUTION

To serve as a "proof of concept," the program listed in Appendix A was written using dBASE III Plus (tm), a relational database programming language. This language lends itself to the structured programming techniques that were used in the development of "SP-1443" (the SP stands for Standard Program), an automated progress payment request generator and contract manager. The program embodies most of the features discussed in the text of this thesis, and is functional to the extent that it will prepare a SF-1443. The program has completed "alpha" testing on simulated data only. The author recommends that it not be placed in general distribution until extensive "beta" testing and additional error trapping have been accomplished.

A. SP-1443 PROGRAM FUNCTIONS AND LIMITATIONS

For the purpose of program development and logical organization, SP-1443 exists in several distinct programs. When the main menu program is executed within dBASE III Plus, the other "programs" are "called" as needed. In practical application, each of these programs would be converted to Procedures to reduce disk access and then compiled to eliminate the need to have the dBASE program resident.

While this program was in the early design phase, it became evident that in order to maintain a high level of data integrity while accommodating the wide range of possible contract activity, it would be necessary to do more than just generate an automated form. It was decided to expand the scope of the program to accomplish these objectives. In so doing, the resulting product more closely resembled a contract management system than a simple forms generator. While the contractor's books and records will still serve as the source documents for the "cost" side of the accounting formula, the "reimbursement" and "changes" aspects of the contract can be traced to actual invoices, payments and contract modifications. The capacity of dBASE III Plus to deal with dates as a data field, make it quite easy to extract a chronological history of the contract at any point in its life.

As the project evolved, it was decided to make the "Contract Reconciliation" one of the menu options available to the user. This function will gather information that was created over a period of time into a simple, concise report that documents the financial progress of the contract. This is another example of expanding competence. A limitation (in a manual system, contract data is usually maintained in several different files, journals, etc.) is acknowledged and "pushed" by the use of an automated tool that has the ability to look at the same data needed to process progress

payments from a different, but also valuable perspective of financial summary.

B. THE RELATIONAL DATABASE

Much has been written on the subject of normalizing data for optimum manipulation. It is not the author's intention to cover that same territory, however, some discussion of the methodology used is in order.

Early database systems used the "flat file" approach in which every conceivable data element would exist in each and every record. Some types of data could be more or less efficiently handled in this manner. If there is a one to one relationship, i.e., a contract has one and only one contract number, one and only one contract date etc., a flat file works well, and there is little waste in the form of duplicated fields. If we compound the problem a little and introduce one to many, or many to many relationships, a flat file can be terribly cumbersome and wasteful. For example, a contract may have many deliverable line items or Contract Line Item Numbers (CLINs). To accurately account for this fact in a flat file, the one to one data (contract number, date, paying office, etc.) would have to be repeated for each different CLIN.

In a relational system, we create a new file that would have the CLIN information and just enough else to uniquely identify it to its parent contract. In this example, we would use the contract number only and ignore the contract

date, paying office, etc. This process forces the designer to more carefully analyze the needs of the system before trying to program it, but the extra effort results in a more powerful and flexible system. As mentioned in Chapter I of this thesis, the SF-1443 contains 32 numeric fields that must be filled out by the contractor. Figure 5 is a screen print of the SP-1443 progress payment request screen. Note that only the essential data for the most current period is requested.

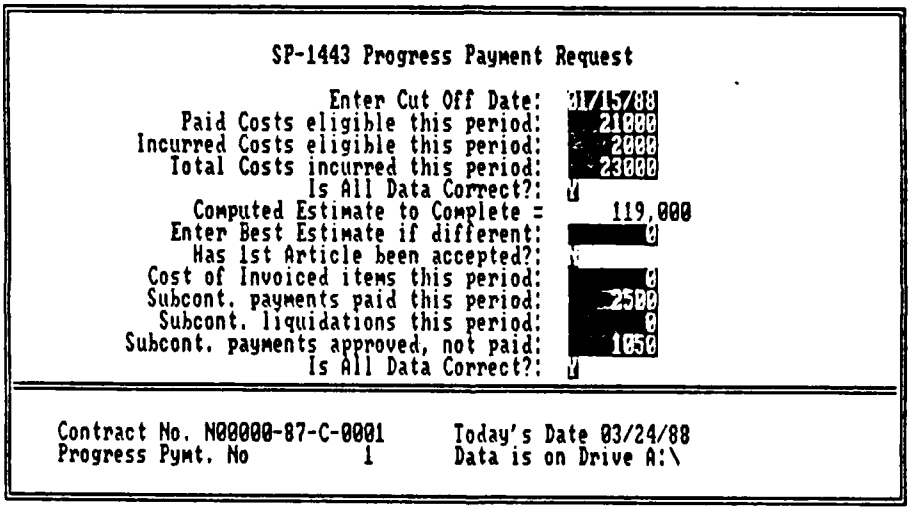


Figure 5. SP-1443 Contract Data

C. STANDARDIZATION

In a 1982 Board of Directors meeting at the Frito-Lay Corporation, Mr. Charles S. Feld, Director of Management Services, presented this single slide. It summarized his philosophy concerning the anticipated growth in information

technclogy following the company's purchase of its first personal computers. [Ref. 4]

Integration is the key to Economics

Control is the key to Integration

Leadership is the key to Control

Vision and Execution are the keys to Leadership

1. Control

These concepts manifest the central theme of this section and the thesis. By offering a timely and effective software equivalent of a Standard Form, the Government can insure standardization through leadership. Control is the operative word. Without it, the Government may find itself at some point in time trying to conform to some commercial standard or (worse yet) several standards for automated data collection and reporting.

2. Consistency

This attribute of standardization takes advantage of the learning curve for the Government as well as contractors. An added advantage of using this type of software is that non-user related types of administrative changes (formulas, cut offs, etc.) can be made to the software without effecting the user. This has the effect of maintaining even greater consistency, and leads to even more effective use of users' time.

3. Efficiency

It was the author's experience in writing SP-1443 that many of the mundane and detail tasks of mathematics and record keeping could be easily relegated to software. This frees the user to concentrate on the content of the reports and to develope the human qualities of judgment and reasonableness. A more competent user is less likely to let a non-math related error get by. In short, using a well designed software aid in contract management more closely matches the attributes and requirements of the task at hand.

D. THOUGHTS ON IMPLEMENTATION

While the author does not hold out SP-1443 as an error free and ready to distribute piece of software, its existence does illustrate the point that such products are indeed possible, and that existing programming languages and personal computers are capable of doing the job quite nicely. To this end, the author recommends the following;

1. Use SP-1443 as a starting point for follow on thesis work,

2. Locate Government contractors and administrative/ paying offices that would be willing to "beta" test the product,

3. Distribute the debugged and tested software free of charge to contractors on a voluntary basis,

4. Investigate the possibility of integrating more contract management functions into the package such as shipping and receiving reports, invoices, delivery reports, project estimation models, DD-250's etc..

E. THE FUTURE OF "STANDARD PROGRAMS"

For over a century, the Federal government has been researching, designing, testing, printing and distributing forms. The last decade has seen a revolution in the office automation area with the introduction of the personal computer. The potential of these machines is scarcely being tapped in the high dollar world of government contract management due largely to the lack of appropriate software. By producing Standard Programs with the same zeal applied to Standard Forms, the author believes that substantial resources can be saved in both the short and long term. In the current era of austere funding, it is an alterative the Government can ill afford to ignore.

V. SUMMARY

Not every agency has the same problems with forms in general or progress payments in particular. Using a microcomputer with an electronic spreadsheet to verify accuracy may be all the automation a company with a resident PRO needs. The sudden recent achievements in computers and data communication have prompted many ADP gurus to "fix" problems that may not warrant fixing. A sense of perspective is certainly an essential tool in any attempt to change or automate any contract management system.

If automation is selected, there are a number of things for the manager to consider. Standardization is perhaps one of the most important. One of the initial goals of DLA was to present "one face to industry." Automation in the Acquisition and Contracting field is an ideal place to put that concept into practice. Standardization means that government employees should not have to learn multiple computer systems just to perform their jobs. It means that companies should not have to change the way they submit progress payment requests depending on the buying activity. It also means that as changes come along, there is just one system to maintain.

To this end, perhaps the government should consider developing and distributing software as well as forms. If

the government provided progress payment software to the contractors who wanted it, standards and compatibility would be assured, and more uniform results could be obtained. The cost of developing or procuring this software would soon be offset by administrative time savings. If the progress payment system works, then expand to include DD 250 receipt inspection automation, or any one of a dozen currently manual contract management systems in a similar manner. If the government fails to take the initiative in this area, an contractor or consultant will emerge that will do the innovation (at the possible expense of the taxpayer). In that event the government may not be able to set and maintain standards to the extent they would like.

In the long view, automation of such repetitive (and dull) tasks like progress payment request preparation, review, validation and payment will be inevitable. As more and more government and industry contract administrators become computer literate, the demand for such automation will build. The only real questions are who will be the innovator and when will it happen. Government has already asserted its considerable influence as a consumer in the marketplace to develope and implement a "standard" programming language, Ada. It is the author's opinion that the same people who brought us the Standard Form (the Federal Government), should bring us the Standard Program, and make it available at little or no cost to contractors

and government agencies alike. The simple fact is that mistakes in form preparation are costly to the government as well as contractors. Any method that will reduce these errors, either systemically or procedurally, should be looked on with great interest by all concerned.

APPENDIX A

SP-1443 SOURCE CODE LISTING

```
* main.prg
* Author: Walt Harsch
* Purpose: Main Program, sets up menu functions
* Calls: base.prg
*        k.prg
*        ppr.prg
*        isp.prg
*        mods.prg
*        recon.prg
*        ktr.prg
*        init.prg
*
* Called by: None
* Input/Output Files: None
*

clear
set talk off
set bell off
store "none selected" to sysk
store "***" to sysr
store date() to sysdate
use conf
store drive to syspath
do base
do while .t.
  store " " to choice
  @ 3,31  say [SP-1443 Main Menu]
  @ 5,20  say [Code:          Function:]
  @ 7,22  say [1            Input/Edit/Terminate a Contract]
  @ 8,22  say [2            Generate a Progress Payment]
  @ 9,22  say [3            Process Invoices/Shipments/Payments]
  @ 10,22 say [4            Process Contract Modifications]
  @ 11,22 say [5            Generate Contract Reconciliation]
  @ 12,22 say [6            Input/Edit Contractor Data]
  @ 13,22 say [7            Set Program Defaults]
  @ 14,22 say [9            Exit and return to DOS]
  @ 16,22 say [Enter your selection: ] get choice
  read
  clear gets
  do case
    case choice = '1'
       do k.prg
    case choice = '2'
```

```
      do ppr.prg
    case choice = '3'
      do isp.prg
    case choice = '4'
      do mod.prg
    case choice = '5'
      do recon.prg
    case choice = '6'
      do ktr.prg
    case choice = '7'
      do init.prg
    case choice = '9'
      clear
      release all
      close databases
      exit
  endcase
  do base
enddo

* eof main.prg^Z* base.prg
* Author: Walt Harsch
* Purpose: To create the basic Input/Output screen for the
* SP-1443 Program.
* Calls: Nothing
* Is Called by: Main.prg, and all functional modules
* Input/Output files: None
*
*
*

clear
set status off


    @ 1,0 to 23,79 double
    @ 18,1 to 18, 78 double
    @ 20,5  say [Contract No. ]
    @ 20,40 say [Today's Date]
    @ 21,5  say [Progress Pymt. No ]
    @ 21,40 say [Data is on Drive ]
    set color to W+
    if sysk="no"
      @ 20,18 say sysk
    else
      @ 20,18 say sysk picture "@R !!!!!!-!!-!-!!!! !!!!"
    endif
    @ 20,53 say sysdate
    @ 21,23 say sysr
    @ 21,57 say syspath
    set color to W
```

```
return

* eof base.prg^Z* k.prg
* Author: Walt Harsch
* Purpose: To Input/Edit/Terminate a contract
* Calls:
* Is called by: Main.prg
* Input/Output Files:
*           K.dbf
*           CLIN.dbf
*           ACO.dbf
*           PO.dbf
*
*
*

clear
store space (17) to contract
mtotal = 0
store .f. to mkans
store " " to choice
do base
@  3,29 say [SP-1443 Contract Data]
@  8,12 say [  Enter Contract Number: ] get contract picture "@R
!!!!!!-!!-!-!!!! !!!!"
read
  if contract = " "
    close databases
    release all like m*
    return
  endif
@ 11,20 say [1     Add a New Contract]
@ 12,20 say [2     Edit a Contract]
@ 13,20 say [3     Delete a Closed Contract]
@ 14,20 say [9     Return to Main Menu]
@ 16,20 say [Enter your selection: ] get choice
read
clear gets
do case

**********************
*     ADD NEW K      *
**********************
   case choice = '1'
**********************
select 1
use ka index ka
restore from ka additive
seek contract
if found()
     do base
     @ 8,12 say [This contract already exists ... Press any key
```

```
to continue]
     minkey = 0
     do while minkey = 0
          minkey = inkey()
     enddo
     release all like m*
     close databases
     return
endif
sysk = contract
do base
store .t. to mansa
store .f. to mansb
store .f. to kkfms
do while .not. mansb
  @  3,30 say [SP-1443 Contract Data]
     @  7,12 say [      Date of Contract: ] get mkdate
  @  8,12 say [                 FMS ?: ] get kkfms picture "Y"
  @  9,12 say [        Subcontracts ?: ] get mksubk picture "Y"
     read
     @ 10,12 say [  Is All Data Correct?: ] get mansb picture 'Y'

     read
     clear gets
enddo
append blank
replace knum with contract
replace kdate with mkdate
replace ksubk with mksubk
replace kfms with kkfms
replace kflag with mkflag
release all like M*
************************
select 2
use kb index kb
restore from kb additive
store .t. to mansa
store .f. to mansb
do while mansa
     do while .not. mansb
     @ 12,12 say [ First Article $ limit: ] get mk1art picture
'99,999,999'
     @ 13,12 say [ Progress Payment Rate: ] get mkppr picture
"999.9"
     @ 14,12 say [       Liquidation Rate: ] get mkpplr picture
"999.9"
     if kkfms
               @ 15,12 say [    Enter Country Code: ] get mkcntry
picture '!!'
          endif
     read
     @ 16,12 say [  Is All Data Correct?: ] get mansb picture "Y"
```

```
      read
      clear gets
   enddo
      if kkfms
           store .f. to mansb
           store .f. to mansa
           @  18,12  say  [  More  Country  Code(s)?:  ]  get  mansa
picture 'Y'
           read
           clear gets
      else
           store .f. to mansa
      endif
      store .t. to mkflag
      append blank
      replace knum with contract
      replace k1art with mk1art
      replace kppr with mkppr
      replace kpplr with mkpplr
      replace kpricei with mkpricei
      replace kpricec with mkpricei
      replace kcntry with mkcntry
      replace kflag with mkflag
enddo
release all like m*
************************
select 3
use aco
restore from aco additive
store .f. to mkans
do base
@  3,30 say [SP-1443 ACO Data]
do while .not. mkans
  @   7,12 say [      Admin. Contracting Office: ] get maconame
picture 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @   8,12 say [                  ACO Name & Title: ] get macotitl
picture 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @   9,12 say [                           ACO Address: ] get macoaddr
picture 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @ 10,12 say [                                ACO City: ] get macocty
picture 'XXXXXXXXXXXXXXXXXX'
  @ 11,12 say [                               ACO State: ] get macost
picture '!!'
  @ 12,12 say [                                ACO Zip: ] get macozip
picture '@R NNNNN-NNNN'
  @ 13,12 say [ Contracting Office I.D. Code: ] get macocd
picture 'NNNNNN'
  @ 14,12 say [    Contracting Office Telex #: ] get macotelex
picture 'NNNNNNNNNNN'
  read
  @ 16,12 say [         Is All Data Correct?: ] get mkans picture
```

```
'Y'
  read
  clear gets
enddo
append blank
replace acoknum with contract
replace aconame with maconame
replace acotitl with macotitl
replace acoaddr with macoaddr
replace acocty with macocty
replace acost with macost
replace acozip with macozip
replace acocd with macocd
replace acotelex with macotelex
release all like m*
************************
select 4
use po
restore from po additive
store .f. to mkans
do base
  @  3,30 say [SP-1443 Pay Office Data]
  do while .not. mkans
  @   7,12  say [     Paying Office Name: ] get mpoadr1 picture
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @   8,12  say [Paying Office Address: ] get mpoadr2 picture
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @   9,12  say [     Paying Office City: ] get mpocty picture
'XXXXXXXXXXXXXXXXXXX'
  @ 10,12 say [  Paying Office State: ] get mpost picture '!!'
  @ 11,12 say [     Paying Office Zip: ] get mpozip picture '@R
NNNNN-NNNN'
  @  12,12  say [     Paying Office Code: ] get mpocd picture
'NNNNNN'
  @  13,12  say [Paying Office Telex #: ] get mpotelex picture
'NNNNNNNNNNNN'
  read
  @ 15,12 say [ Is All Data Correct?: ] get mkans picture 'Y'
  read
  clear gets
enddo
append blank
replace poknum with contract
replace poadr1 with mpoadr1
replace poadr2 with mpoadr2
replace pocty with mpocty
replace post with mpost
replace pozip with mpozip
replace pocd with mpocd
replace potelex with mpotelex
release all like m*
************************
```

```
select 5
use clin index clin
store .t. to mkansa
mgtotal = 0
do while mkansa
     do base
     @  3,30 say [SP-1443 CLIN Data]
  restore from clin additive
     store .f. to mkansb
  do while .not. mkansb
          @   7,12 say [                    CLIN #: ] get mclin
picture '!!!!!!!'
          seek contract+mclin
          if found()
               @  9,12 say [You have entered a duplicate CLIN -]
               @ 10,12 say [Press any key to continue...]
               minkey = 0
               do while minkey = 0
                    minkey = inkey()
               enddo
               loop
          endif
          mtotal = 0
          @   8,12 say [                  Quantity: ] get mcqtyi
picture '9999999'
     @  9,12 say [         Unit of Issue: ] get mcui picture '!!!'

     @ 10,12 say [             Unit Price: ] get mcpricei picture
'999999.99'
     @ 11,12 say [         Delivery Date: ] get mcduei
     if kkfms
     @ 12,12 say [           Country Code: ] get mccntry picture
'!!'
     endif
     read
     @ 14,12 say [   Is All Data Correct?: ] get mkansb picture
'Y'
     read
     clear gets
     enddo
  append blank
  replace cknum with contract
  replace clin with mclin
  replace cqtyi with mcqtyi
     replace cqtyc with mcqtyi
  replace cui with mcui
  replace cpricei with mcpricei
     replace cpricec with mcpricei
  replace cduei with mcduei
     replace cduec with mcduei
  replace ccntry with mccntry
     replace cqtysh with 0
```

```
      replace cqtyin with 0
      store .f. to mkansa
   @ 16,12 say [Are There More CLIN's?: ] get mkansa picture 'Y'
   read
   clear gets
      mtotal = mtotal + (mcqtyi*mcpricei)
      mgtotal = mgtotal + mtotal
      ************************
      select 2
      seek contract+mccntry
      replace kpricei with kpricei+mtotal
      replace kpricec with kpricec+mtotal
      ************************
      select 1
      seek contract
      replace kgtotal with mgtotal
   ************************
      select 5
enddo
use
do base
@ 10,12 say [Total price for this contract is $: ]
@ 10,48 say mgtotal picture '99,999,999.99'
@ 12,12 say [Press any key to continue ...]
minkey = 0
do while minkey = 0
     minkey = inkey()
enddo
release all like m*


************************
*        EDIT          *
************************
case choice = '2'
************************

select 1
use ka index ka
seek contract
if eof ()
     do base
     @    7,12  say  [Contract  not  found  ...  Press  any  key  to
continue]
     minkey = 0
     do whole minkey =0
          minkey = inkey()
     enddo
     loop
endif
store kfms to kkfms
store 'US' to mkcntry
```

```
if kkfms
     @  8,12 say [Enter Country Code: ] get mkcntry picture "!!"
     read
endif
store kflag to mkflag
store kdate to mkdate
store ksubk to mksubk
if .not. mkflag
     do base
     @  7,12 say [This contract has already been edited once. ]
     @  8,12 say [Use the Contract Modification option from the
Main Menu]
     @  9,12 say [ ... Press any key to continue]
     minkey = 0
     do while minkey = 0
          minkey = inkey()
     enddo
     loop
endif

sysk = contract
do base
store .t. to mansa
store .f. to mansb
kkfms = mkfms
do while .not. mansb
  @  3,30 say [SP-1443 Contract Data]
     @  7,12 say [      Date of Contract: ] get mkdate
  @  8,12 say [                FMS ?: ] get kkfms picture "Y"
  @  9,12 say [        Subcontracts ?: ] get mksubk picture "Y"
     read
     @ 10,12 say [  Is All Data Correct?: ] get mansb picture 'Y'

     read
     clear gets
enddo
replace knum with contract
replace kdate with mkdate
replace ksubk with mksubk
replace kfms with kkfms
replace kflag with .f.
release all like M*
************************
select 2
use kb index kb
restore from kb additive
store klart to mklart
store kcntry to mkcntry
store kppr to mkppr
store kpplr to mkpplr
store kpricei to mkpricei
store .f. to mansa
```

```
do while mansa
     do while .not. mansb
     @ 12,12 say [ First Article $ limit: ] get mk1art picture
'99,999,999'
     @ 13,12 say [ Progress Payment Rate: ] get mkppr picture
"999.9"
     @ 14,12 say [       Liquidation Rate: ] get mkpplr picture
"999.9"
     if kkfms
               @ 15,12 say [    Enter Country Code: ] get mkcntry
picture '!!'
          endif
     read
     @ 16,12 say [  Is All Data Correct?: ] get mansb picture "Y"

     read
     clear gets
  enddo
     if kkfms
          store .f. to mansb
          store .f. to mansa
          @  18,12  say  [  More  Country  Code(s)?:  ]  get  mansa
picture 'Y'
          read
          clear gets
     else
          store .f. to mansa
     store .f. to mkflag
     replace knum with contract
     replace k1art with mk1art
     replace kppr with mkppr
     replace kpplr with mkpplr
     replace kpricei with mkpricei
     replace kpricec with mkpricei
     replace kcntry with mkcntry
     replace kflag with mkflag
enddo
release all like m*
************************
select 3
use aco index aco
store aconame to maconame
store acotitl to macotitl
store acoaddr to macoaddr
store acocty to macocty
store acost to macost
store acozip to macozip
store acocd to macocd
store acotelex to macotelex
store .f. to mkans
do base
@  3,30 say [SP-1443 ACO Data]
```

```
do while .not. mkans
     @  7,12 say [     Admin. Contracting Office: ] get maconame
picture 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @   8,12 say [                 ACO Name & Title: ] get macotitl
picture 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @   9,12 say [                          ACO Address: ] get macoaddr
picture 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @ 10,12 say [                              ACO City: ] get macocty
picture 'XXXXXXXXXXXXXXXXXX'
  @ 11,12 say [                             ACO State: ] get macost
picture '!!'
  @ 12,12 say [                              ACO Zip: ] get macozip
picture '@R NNNNN-NNNN'
  @ 13,12 say [ Contracting Office I.D. Code: ] get macocd
picture 'NNNNNN'
  @ 14,12 say [    Contracting Office Telex #: ] get macotelex
picture 'NNNNNNNNNNNN'
  read
  @ 16,12 say [         Is All Data Correct?: ] get mkans picture
'Y'
  read
  clear gets
enddo
replace acoknum with contract
replace aconame with maconame
replace acotitl with macotitl
replace acoaddr with macoaddr
replace acocty with macocty
replace acost with macost
replace acozip with macozip
replace acocd with macocd
replace acotelex with macotelex
release all like m*
************************
select 5
use po index po
seek contract
store poadr1 to mpoadr1
store poadr2 to mpoadr2
store pocty to mpocty
store post to mpost
store pozip to mpozip
store pocd to mpocd
store potelex to mpotelex
store .f. to mkans
do base
@  3,30 say [SP-1443 Pay Office Data]
do while .not. mkans
  @   7,12 say [     Paying Office Name: ] get mpoadr1 picture
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @   8,12 say [Paying Office Address: ] get mpoadr2 picture
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
```

```
  @   9,12 say [     Paying Office City: ] get mpocty picture
'XXXXXXXXXXXXXXXXXXXX'
  @ 10,12 say [  Paying Office State: ] get mpost picture '!!'
  @ 11,12 say [     Paying Office Zip: ] get mpozip picture '@R
99999-9999'
  @ 12,12 say [     Paying Office Code: ] get mpocd picture
'NNNNNN'
  @ 13,12 say [Paying Office Telex #: ] get mpotelex picture
'NNNNNNNNNNNN'
  read
  @ 15,12 say [ Is All Data Correct?: ] get mkans picture 'Y'
  read
  clear gets
enddo
replace poknum with contract
replace poadr1 with mpoadr1
replace poadr2 with mpoadr2
replace pocty with mpocty
replace post with mpost
replace pozip with mpozip
replace pocd with mpocd
replace potelex with mpotelex
release all like m*
************************
select 5
use clin
store .t. to mkansa
store space(7) to mclin
locate for cknum=contract
do while mkansa
     do base
     @  3,30 say [SP-1443 CLIN Data]
     store .f. to mkansb
  do while .not. mkansb
          if .not. found()
               @  9,12 say [No CLINs found -]
               @ 10,12 say [Press any key to continue...]
               minkey = 0
               do while minkey = 0
                    minkey = inkey()
               enddo
               do base
               return
          endif
          store clin to mclin
          store cqtyi to mcqtyi
          store cpricei to mcpricei
          store cduei to mcduei
          store ccntry to mccntry
          store cui to mcui
          mtotal = mcqtyi * mcpricei
          *************************
```

```
          select 2
          seek contract+mccntry
          replace kpricei with kpricei-mtotal
          replace kpricec with kpricei-mtotal
          release all like m*
          *************************
          select 5
          @   7,12 say [                    CLIN #: ] get mclin
picture '!!!!!!!'
          @   8,12 say [                 Quantity: ] get mcqtyi
picture '9999999'
     @  9,12 say [         Unit of Issue: ] get mcui picture '!!!'

     @ 10,12 say [             Unit Price: ] get mcpricei picture
'999999.99'
     @ 11,12 say [         Delivery Date: ] get mcduei
     if kkfms
     @ 12,12 say [           Country Code: ] get mccntry picture
'!!!'
     endif
     read
     @ 14,12 say [   Is All Data Correct?: ] get mkansb picture
'Y'
     read
     enddo
  replace cknum with contract
  replace clin with mclin
  replace cqtyi with mcqtyi
     replace cqtyc with mcqtyi
  replace cui with mcui
  replace cpricei with mcpricei
     replace cpricec with mcpricei
  replace cduei with mcduei
     replace cduec with mcduei
  replace ccntry with mccntry
     replace cqtysh with 0
     replace cqtyin with 0
     mtotal = mcqtyi * mcpricei
     mgtotal = mgtotal + mtotal
     *************************
     select 2
     seek contract+mccntry
     replace kpricei with kpricei+mtotal
     replace kpricec with kpricei+mtotal
     *************************
     continue
     if found()
          @ 16,12 say [Press any key for next CLIN]
          minkey = 0
          do while minkey = 0
               minkey = inkey()
          enddo
```

```
     else
          store .f. to mkansa
     endif
     clear gets
enddo
*************************
select 1
seek contract
replace kgtotal with mgtotal
*************************
use
do base
@ 10,12 say [Total price for this contract is $: ]
@ 10,48 say mgtotal picture '99,999,999.99'
@ 12,12 say [Press any key to continue ...]
minkey = 0
do while minkey = 0
     minkey = inkey()
enddo
release all like m*

************************
*       DELETE         *
************************
case choice = '3'

************************

store .f. to mansb
@ 16,20 say [Are You Shure you wish to delete this Contract?: ]
get mansb picture 'Y'
read
if .not. mansb
  loop
endif
use ka index ka
delete while knum=contract
pack
use kb index kb
delete while knum=contract
pack
use inv index inv
delete while iknum=contract
pack
use mod index mod
delete while moknum=contract
pack
use ppr index ppr
delete while pknum=contract
pack
use po index po
delete while poknum=contract
```

```
pack
use clin index clin
delete while cknum=contract
pack
use ship index ship
delete while sknum=contract
pack
use aco index aco
delete while acoknum=contract
pack
use pay index pay
delete while pyknum=contract
pack
use modr index modr
delete while moknum=contract
pack
use invr index invr
delete while iknum=contract
pack
use shipr index shipr
delete while sknum=contract
pack
release all like m*
otherwise
  close database
     clear
  sysk= [none selected]
  return
endcase
close database
clear
sysk = [none selected]
return
^Z
* init.prg
* Author: Walt Harsch
* Purpose: To set program default drive and to create the "mem"
*   files if not already done.
* Calls:
* Is called by: Main.Prg
*
*
*
clear
do base
store " " to choice
*
@  3,31 say [SP-1443 Program Setup]
@  8,15 say [1.   Set Up Data File Default Drive]
@  9,15 say [2.   Select Printer]
@ 10,15 say [9.   Return to Main Menu]
@ 12,20 say [Enter Your Selection: ] get choice
```

```
read
clear gets
do case
     case choice = '1'
          clear
          do base
          use conf
          store drive to mdrive
          @ 3,31  say [SP-1443 Program Setup]
     @ 11,22 say [Drive & Path              : ] get mdrive picture
"!!!!!!!!!!!!!"
     read
     clear gets
     set default to &mdrive
     syspath = mdrive
          append blank
          replace drive with mdrive
     release mdrive
     case choice = '2'
          do base
          mprint = 0
          @ 3,31 say [SP-1443 Printer Select]
          @ 7,20 say [1.  Epson]
          @ 8,20 say [2.  MT-160]
          @ 9,20 say [3.  H.P. LaserJet]
          @ 11,20 say [Select one: ] get mprint picture '9'
          read
          clear gets
          do case
               case mprint = 1
                    store "epson" to mcode
               case mprint = 2
                    store "mt180" to mcode
               case mprint = 3
                    store "laser" to mcode
     endcase
*
     if .not. file(ka.mem)
               use ka
               store space(17) to mknum           &&   contract
number
               store ctod("00/00/00") to mkdate   &&   contract
date
        store .f. to mksubk                       &&  subcontract
flag
               store .f. to mkfms                      && Foreign
Milit. Sales flag
               store .f. to mkflag                           &   &
file lock flag
               store 0 to mkgtotal                     && initial
grand total of contract
               save all like m* to ka
```

```
                release all like m*
     endif
     if .not. file(kb.mem)
                use kb
                store space(17) to mknum        && contract number
                store 0 to mk1art               &&  1st  article  $
limit
                store "US" to mkcntry           && Country code
                store 0 to mkppr                     &&  progress
payment rate
                store 0 to mkpplr               &&  prog.  pay.
liquidation rate
                store 0 to mkduek               && $ due on contract

                store 0 to mkdisc               &&  total  doscounts
taken
                store 0 to mkpricei                  &&  initial
contract price
                store 0 to mkpricec                  &&  current
price
                store 0 to mkpectd              &&  paid  eligible
cost to date
                store 0 to mkiectd              && incurred eligible
cost to date
                store 0 to mktcitd              && tot.cost incurred
to date
                store 0 to mkskptd              && sub contract paid
to date
                store 0 to mkskltd              &&  sub  contract
liquidated to date
                store 0 to mkskwait                  &&  sub  K
awaiting payment
                store 0 to mkdctd               && delivered cost to
date
                store 0 to mkinvtd              &&  total  invoiced
amount to date
                store 0 to mkpptd               &&  total  progress
payment to date
                store 0 to mkinvpd              &&  total  invoices
paid
                store 0 to mktciv               &&  total  cost
invoiced
                store .f. to mkflag                  &&  file  lock
flag
                save all like m* to kb
                release all like m*
     endif
     if .not. file(ship.mem)
                use ship
                store space(7) to mshipno
                store ctod("00/00/00") to mshipdt
                store space(17) to mknum
```

```
                save all like m* to ship
                release all like m*
        endif
        if .not. file(inv.mem)
                use inv
                store space(7) to minvono
                store ctod("00/00/00") to minvodt
                store 0 to minvamt
                store space(17) to mknum
                save all like m* to inv.mem
                release all like m*
        endif
        if .not. file(clin.mem)
                use clin
                store space(7) to mclin
                store 0 to mcqtyi
                store 0 to mcqtyc
                store 0 to mcpricei
                store 0 to mcpricec
                store ctod("00/00/00") to mcduei
                store ctod("00/00/00") to mcduec
                store [US] to mccntry                   && country code

                store 0 to mcqtysh                      && qty shipped
                store 0 to mcqtyin                      && qty invoived

                store [EA] to mcui                      && unit of
issue
                store space(17) to mknum                && contract
number
                store space(25) to mdesc                && description
                save all like m* to clin
                release all like m*
        endif
        if .not. file(mod.mem)
                use mod
                store space(6) to mmodno
                store ctod("00/00/00") to mmoddt
                store 0 to modamt
                store space(17) to mmoknum
                store .f. to mmoadmin
                store .f. to mmoflag
                save all like m* to mod
                release all like m*
        endif
        if .not. file(modr.mem)
                use modr
                store space(17) to mmoknum
                store space(6) to mmodno
                store space(7) to mmoclin
                store 0 to mmoqtyf
                store 0 to mmoqtyt
```

```
                store 0 to mmoprf
                store 0 to mmoprt
                store ctod("00/00/00") to mmoduef
                store ctod("00/00/00") to mmoduet
                store .f. to mmoadmin
                store .f. to mmoflag
                save all like m* to modr
                release all like m*
        endif
        if .not. file(ppr.mem)
                use ppr
                store ctod("00/00/00") to mpcod
                store 0 to mprno
                store ctod("00/00/00") to mpprdt
                store 0 to mpramt
                store 0 to mpdcst
                store 0 to mpincst
                store 0 to mpskppd
                store 0 to mpsklpd
                store 0 to mpskwait
                store 0 to mpcntry
                store 0 to mptcst
                store 0 to mpciv
                store space(17) to mknum
                store .f. to mpflag
                save all like m* to ppr.mem
                release all like m*
        endif
        if .not. file(aco.mem)
                use aco
                store space(30) to maconame        && Contract
Admin. Office
                store space(30) to macotitl        && ACO name +
title
                store space(30) to macoaddr
                store space(20) to macocty
                store space(2) to macost
                store space(9) to macozip
                store space(6) to macocd
                store space(13) to macotelex
                store space(17) to mknum
                save all like m* to aco.mem
                release all like m*
        endif
        if .not. file(po.mem)
                use po
                store space(30) to mpoadr1
                store space(30) to mpoadr2
                store space(20) to mpocty
                store space(2) to mpost
                store space(9) to mpozip
                store space(6) to mpocd
```

```
                store space(13) to mpotelex
                store space(17) to mknum
                save all like m* to po.mem
                release all like m*
        endif
        if .not. file(pay.mem)
                use pay
                store ctod("00/00/00") to mpydate
                store 0 to mpyamt
                store 0 to mpyckno
                store 0 to mpydisc
                store space(17) to mknum
                save all like m* to pay
                release all like m*
        endif
        if .not. file(epson.mem)
                store [chr(15)] to mc
                store [chr(27)+"0"] to m81
                store [chr(27)+"G"] to md
                store [chr(27)+"H"] to ms
                store [chr(27)+"@"] to mr
                store [chr(27)+"8"] to ml
                store [chr(13)] to mcr
                store [chr(12)] to mff
                save all like m* to epson
                release all like m*
        endif
        if .not. file(mt180.mem)
                store [chr(15)] to mc
                store [chr(27)+"0"] to m81
                store [chr(27)+"E"] to md
                store [chr(27)+"F"] to ms
                store [chr(27)+"@"] to mr
                store [chr(27)+"8"] to ml
                store [chr(13)] to mcr
                store [chr(12)] to mff
                save all like m* to mt180
                release all like m*
        endif
*       if .not. file(laser.mem)
*               store [chr  (laser printer control codes here)
* endif
        if .not. file(ktr.mem)
                use ktr
                store space(30) to mcname
                store space(30) to mcaddr
                store space(20) to mccity
                store space(2) to mcst
                store space(9) to mczip
                store space(5) to mcfscm
                store .f. to mcsmall
                store space(13) to mctelex
```

```
                save all like m* to ktr
                release all like m*
        endif
             case choice = '9'
             release choice
             do base
             return
endcase
release choice
release all like M*
release all like k*
close databases
clear
do base
return

*eof init.prg
^Z* ktr.prg
* Author: Walt Harsch
* Purpose: To create/edit the Contractor's attribute
*   information and maintain the appropriate disk
*   file.
* Calls: None
* Is called by: Main.prg
* Input/Output files: ktr.dbf
*
*

clear

do base

use ktr
restore from ktr additive
store .f. to mkans
do while .not. mkans
  @ 3,29  say [SP-1443 Contractor Data]
  @ 7,12  say [     Contractor Name: ] get mcname
  @ 8,12  say [             Address: ] get mcaddr
  @ 9,12  say [                City: ] get mccity
  @ 10,12 say [               State: ] get mcst picture '!!'
  @ 11,12 say [                 Zip: ] get mczip picture '@R
99999-9999'
  @ 12,12 say [    Small Business ?: ] get mcsmall picture 'Y'
  @ 13,12 say [           FSCM Code: ] get mcfscm picture 'NNNNN'

  @ 14,12 say [      TELEX, Easylink: ] get mctelex picture '@B
9999999999999'
  read
     @ 16,12 say [Is All Data Correct?: ] get mkans picture 'Y'
     read
     clear gets
```

```
enddo
append blank
replace cname with mcname
replace caddr with mcaddr
replace ccity with mccity
replace cst with mcst
replace czip with mczip
replace csmall with mcsmall
replace cfscm with mcfscm
replace ctelex with mctelex
close databases
release all like m*

clear
do base
return

* eof ktr.prg
^Z
* ppr.prg
* Author : Walt Harsch
* Purpose: to collect progress payment raw input from user
*   and compute, using historical data, a new progress payment
*   request
*
*
*
*
************************
select 1
store space(17) to contract
do base
store .t. to mans
do while mans
     @  3,24 say [SP-1443 Progress Payment Request]
     @ 10,9  say  [Enter contract number (or CR to exit): ] get
contract picture '@R !!!!!!-!!-!-!!!! !!!!'
     read
     if contract = " "
          return
     endif
     use ka index ka
     restore from ka additive
     seek contract
     if eof()
          @ 12,12 say [Contract Not Found . . . Press any key to
try again]
          minkey = 0
          do while minkey = 0
               minkey = inkey()
          enddo
          clear
```

```
          do base
          loop
     endif
     mans = .f.
enddo
store kfms to kkfms
store kdate to mkdate
store ksubk to mksubk
store "US" to mtfms && think abiout this one.
if kkfms
     @ 12,12 say [Enter Country Code: ] get mtfms picture '!!'
     read
endif
clear gets
************************
select 2
use kb index kb
restore from kb additive
clear
seek contract+mtfms
store klart to mklart
store kcntry to mcntry
store kppr to mkppr
store kpplr to mkpplr
store kduek to mkduek
store kpricei to mkpricei
store kpricec to mkpricec
store kpectd to mkpectd
store kiectd to mkiectd
store ktcitd to mktcitd
store kskptd to mkskptd
store kskltd to mkskltd
store kskwait to mkskwait
store kdctd to mkdctd
store ktciv to mktciv
store kinvtd to mkinvtd
store kpptd to mkpptd
store kflag to mkflag
*****************************
select 3
use ktr
restore from ktr additive
store cname to mcname
store caddr to mcaddr
store ccity to mccity
store cst to mcst
store czip to mczip
store cfscm to mcfscm
store ctelex to mctelex
store csmall to mcsmall
*****************************
select 4
```

```
use ppr index ppr
restore from ppr additive
set filter to pknum=contract
seek contract+mkcntry

mprno = mprno + 1
sysr = mprno
sysk = contract
clear
do base

if mklart = 0
     store .t. to mart
else
     store .f. to mart
endif
store .f. to mans
do while .not. mans
     @    3,24 say [SP-1443 Progress Payment Request]
     @     5,12 say [                    Enter Cut Off Date: ] get
mpcod
     if mcsmall
          @ 6,12 say [Incurred Costs eligible this period: ] get
mpincst picture '99999999'
     else
          @ 6,12 say [    Paid Costs eligible this period: ] get
mpdcst picture '99999999'
          @ 7,12 say [Incurred Costs eligible this period: ] get
mpincst picture '99999999'
     endif
     @     8,12 say [   Total Costs incurred this period: ] get
mptcst picture '99999999'
     read
     @     9,12 say [                 Is All Data Correct?: ] get
mans picture 'Y'
     read
     if .not. mans
          loop
     endif
     mest = mkpricec - (mktcitd + mptcst)
     mnest = 0
     @   10,12 say [     Computed Estimate to Complete = ]
     @   10,51 say mest picture '99,999,999'
     @    11,12 say [    Enter Best Estimate if different: ] get
mnest picture '99999999'
     if .not. mart
          @    12,12 say [      Has 1st Article been accepted?: ]
get mart picture 'Y'
     endif
     if mnest > 0
          mest = mnest
     endif
```

```
     @    13,12 say [ Cost of Invoiced items this period: ] get
mpciv picture '99999999'
     if mksubk
          @ 14,12 say [ Subcont. payments paid this period: ] get
mpskppd picture '99999999'
          @ 15,12 say [  Subcont. liquidations this period: ] get
mpsklpd picture '99999999'
          @ 16,11 say [Subcont. payments approved, not paid: ]
get mpskwait picture '99999999'
     endif
     read
     @    17,12 say [                  Is All Data Correct?: ] get
mans picture 'Y'
     read
     clear gets
enddo
*
*    COMPUTATIONS & TESTS
*
* --- loss check
*
mlr = 1
if mktcitd + mptcst + mest > mkpricec
     clear
     do base
     @  3,30 say [SP-1443 Loss Contract Worksheet]
     @  7,12 say [This contract appears to be in a loss position]

     mextra = 0
     @  9,12 say [Enter pending Change Orders, Unpriced Orders o-
Mods]
     @ 10,12 get mextra picture '99999999'
     mlr = (mkpricec + mextra) / (mktcitd+mptcst+mest)
endif
*
* --- eligible costs <= total costs
*
if max(mpincst,mpdcst) > mptcst
     clear
     do base
     @ 3,30 say [SP-1443 Logical Error]
     @ 12,12 say [Eligible Costs cannot be greater than Total
Costs]
     @ 14,12 say [Press any key to start over]
     minkey = 0
     do while minkey = 0
          minkey = inkey()
     enddo
     *return to PPR main screen here
endif

if mcsmall
```

```
     ln9  = 0
     ln10 = mkiectd + mpincst
else
     ln9  = mkpectd + mpdcst
     ln10 = mkiectd + mpincst
endif
if int(ln9) < ln9
     ln9 = int(ln9) + 1
endif
if int(ln10) < ln10
     ln10 = int(ln10) + 1
endif
ln11      = (ln9 + ln10) * mlr
if int(ln11) < ln11
     ln11 = int(ln11) + 1
endif
ln12a     = mktcitd + mptcst
if int(ln12a) < ln12a
     ln12a = int(ln12a) + 1
endif
ln12b     = mest
if int(ln12b) < ln12b
     ln12b = int(ln12b) + 1
endif
ln13      = ln11 * (mkppr/100)
if int(ln13) < ln13
     ln13 = int(ln13) + 1
endif
if mksubk
     ln14a          = mkskptd + mpskppd
     if int(ln14a) < ln14a
          ln14a = int(ln14a) + 1
     endif
  ln14b        = mkskltd + mpsklpd
     if int(ln14b) < ln14b
          ln14b = int(ln14b) + 1
     endif
  ln14c        = ln14a - ln14b
     ln14d          = mpskwait
     if int(ln14d) < ln14d
          ln14d = int(ln14d) + 1
     endif
     ln14e          = ln14c + ln14d
else
     ln14a          = 0
     ln14b          = 0
     ln14c          = 0
     ln14d          = 0
     ln14e          = 0
endif
ln15      = ln13 + ln14e
ln16      = mkpricec * (mkppr/100)
```

```
if int(ln16) < ln16
     ln16 = int(ln16) + 1
endif
ln17       = min( ln15, ln16)
ln18       = mkpptd
ln19       = ln17 - ln18
ln20a      = min(mktciv + mpciv, mkpricec)
ln20b      = ln11 - ln20a
ln20c      = ln20b * (mkppr/100)
if int(ln20c) < ln20c
     ln20c = int(ln20c) + 1
endif
ln20d      = ln14e
ln20e      = ln20c + ln20d
ln21a      = mkinvtd
ln21b      = mkpricec - ln21a
ln21c      = ln21b * (mkppr/100)
if int(ln21c) < ln21c
     ln21c = int(ln21c) + 1
endif
ln21d = 0
ln21e      = ln21c - ln21d
ln22 = min( ln20c, ln21e)
ln23 = mkinvtd * (mkpplr/100)
if int(ln23) < ln23
     ln23 = int(ln23) + 1
endif
ln24 = ln18 - ln23
ln25 = ln22 - ln24
ln26 = min( ln25, ln19)
*
* draft report starts here:
************************
select 5
use aco index aco
restore from aco additive
seek contract
store aconame to maconame
store acotitl to macotitl
store acoaddr to macoaddr
store acocty to macocty
store acost to macost
store acozip to macozip
store acocd to macocd
store acotelex to macotelex
************************
select 6
use po index po
restore from po additive
seek contract
store poadr1 to mpoadr1
store poadr2 to mpoadr2
```

```
store pocty to mpocty
store post to mpost
store pozip to mpozip
store pocd to mpocd
store potelex to mpotelex
*
clear
do base
@  3,30 say [SP-1443 Draft Report]
@  7,12 say [Ready Printer, Press any key to print]
minkey = 0
do while minkey = 0
     minkey = inkey()
enddo
store .t. to mflag
store .t. to mans
restore from epson additive
do while mans
*
*
*
* The following is an unindented do loop
*
*
*
set device to print
@ 1,1 say &mr
@ 1,1 say &m81
@ 1,1 say &mc
@ 1,1 say &ml
@           1     ,     1           s     a     y
[-------------------------------------------------------------
-------------------------------------------------------------
----]
@ 2,1 say       "
CONTRACTOR'S REQUEST FOR PROGRESS PAYMENT
 OMB No. 3090-0105"
@           3     ,     1           s     a     y
[-------------------------------------------------------------
-------------------------------------------------------------
----]
@  4,1  say         "IMPORTANT:  This  form  is  to  be  completed  in
accordance with instructions on reverse.
                        "
@           5     ,     1           s     a     y
[-------------------------------------------------------------
-------------------------------------------------------------
----]
@ 6,1 say      "
SECTION I - IDENTIFICATION INFORMATION
                  "
@ 7,1 say      [ ]
```

```
@ 8,1 say      "1. TO:  NAME AND ADDRESS OF CONTRACTING OFFICE
                 |2.  FROM:    NAME  AND  ADDRESS  OF  CONTRACTOR
(Include ZIP Code)    "
@ 9,1 say      "
                 |
                "
@ 10,1 say     "
                 |
                "
@ 10,17 say &md
@ 10,17 say maconame
@ 10,80 say mcname
@ 10,130 say &ms
@ 11,1 say     "
                 |
                "
@ 11,17 say &md
@ 11,17 say macoaddr
@ 11,80 say mcaddr
@ 11,130 say &ms
@ 12,1 say     "
                 |
                "
@ 12,17 say &md
@ 12,17 say macocty
@ 12,38 say macost
@ 12,41 say macozip picture '@R 99999-9999'
@ 12,80 say mccity
@ 12,101 say mcst
@ 12,104 say mczip picture '@R 99999-9999'
@ 12,130 say &ms
@ 13,1 say     "PAYING OFFICE:

-----------------------------------------------------------------"

@ 14,1 say     "
                 |3.  SMALL  BUSINESS  |4.  CONTRACT  NUMBER
|CONTRACT PRICE     "
@ 14,17 say &md
@ 14,17 say mpoadr1
@ 14,130 say &ms
@ 15,1 say     "
                 |                     |                          |
                "
@ 15,17 say &md
@ 15,17 say mpoadr2
@ 15,130 say &ms
@ 16,1 say     "
                 |  [ ]YES    [ ]NO  |                          |  $
               "
@ 16,2 say &md
@ 16,19 say mpocty
```

```
@ 16,40 say mpost
@ 16,43 say mpozip picture '@R 99999-9999'
if mcsmall
     @ 16,74 say [X]
else
     @ 16,83 say [X]
endif
@ 16,92 say contract picture '@R !!!!!!-!!-!-!!!! !!!!'
@ 16,120 say mkpricec picture '99,999,999'
@ 16,130 say &ms
@           1    7    ,    1           s    a    y
"-------------------------------------------------------------
-------------------------------------------------------------
----"
@ 18,1 say    "              6. RATES              |    7. DATE OF
INITIAL AWARD    |8A. PROGRESS PAYMENT REQUEST |8B. DATE OF THIS
REQUEST         "
@           1    9    ,    1           s    a    y
"-------------------------------------------------------------
----|    NO.                          |
  "
@ 20,1 say    "A. PROG. PYMTS.  |B. LIQUIDATION  |    A. YEAR
|   B. MONTH    |                          |
              "
@ 20,80 say &md
@ 20,82 say mprno picture '999'
if kkfms
     @ 20,87 say mkcntry picture '!!'
endif
@ 20,109 say sysdate
@ 20,130 say &ms
@ 21,1 say    "                  |               |
|               |                        |
              "
@ 21,2 say &md
@ 21,8 say mkppr picture '@R 999.9 %'
@ 21,26 say mkpplr picture '@R 999.9 %'
@ 21,44 say year(mkdate)
@ 21,62 say month(mkdate)
@ 21,130 say &ms
@           2    2    ,    1           s    a    y
"-------------------------------------------------------------
-------------------------------------------------------------
----"
@ 23,1 say    "                                        SECTION II -
STATEMENT OF COSTS UNDER THIS CONTRACT THROUGH
                "
@ 23,95 say &md
@ 23,98 say mpcod
@ 23,130 say &ms
@           2    4    ,    1           s    a    y
"-------------------------------------------------------------
```

```
-----------------------------------------------------------------
----"
@ 25,1 say     " 9. PAID COSTS ELIGIBLE UNDER PROGRESS PAYMENT
CLAUSE
    |$"
@ 25,120 say &md
@ 25,120 say ln9 picture '99,999,999'
@ 25,130 say &ms
@ 26,1 say    "10. INCURRED COSTS ELIGIBLE UNDER PROGRESS PAYMENT
CLAUSE
| "
@ 26,120 say &md
@ 26,120 say ln10 picture '99,999,999'
@ 26,130 say &ms
@ 27,1 say     "11. TOTAL COSTS ELIGIBLE FOR PROGRESS PAYMENTS
(Item 9 plus 10)
|----------------| "
@ 27,120 say &md
@ 27,120 say ln11 picture '99,999,999'
@ 27,130 say &ms
@ 28,1 say    "12a. TOTAL COSTS INCURRED TO DATE
                                                    |$
| "
@ 28,103 say &md
@ 28,103 say ln12a picture '99,999,999'
@ 28,130 say &ms
@ 29,1 say    "  b. ESTIMATED ADDITIONAL COST TO COMPLETE
                                                   |
| "
@ 29,103 say &md
@ 29,103 say ln12b picture '99,999,999'
@ 29,130 say &ms
@ 30,1 say    "13. ITEM 11 MULTIPLIED BY ITEM 6a.

|----------------| "
@ 30,120 say &md
@ 30,120 say ln13 picture '99,999,999'
@ 30,130 say &ms
@ 31,1 say    "14a. PROGRESS PAYMENTS PAID TO SUBCONTRACTORS
                                                   |
| "
@ 31,103 say &md
@ 31,103 say ln14a picture '99,999,999'
@ 31,130 say &ms
@  32,1  say        "    b.  LIQUIDATED  PROGRESS  PAYMENTS  TO
SUBCONTRACTORS                                      |
              | "
@ 32,103 say &md
@ 32,103 say ln14b picture '99,999,999'
@ 32,130 say &ms
@  33,1  say        "    c.  UNLIQUIDATED  PROGRESS  PAYMENTS  TO
SUBCONTRACTORS (Item 14a less 14b)                  |
```

```
            | "
@ 33,103 say &md
@ 33,103 say ln14c picture '99,999,999'
@ 33,130 say &ms
@ 34,1 say     "   d. SUBCONTRACT PROGRESS BILLINGS APPROVED FOR
CURRENT PAYMENT                                                 |
   | "
@ 34,103 say &md
@ 34,103 say ln14d picture '99,999,999'
@ 34,130 say &ms
@ 35,1 say     "   e. ELIGIBLE SUBCONTRACTOR PROGRESS PAYMENTS
(Item 14c plus 14e)
|----------------| "
@ 35,120 say &md
@ 35,120 say ln14e picture '99,999,999'
@ 35,130 say &ms
@ 36,1 say    "15. TOTAL DOLLAR AMOUNT (Item 13 plus 14e)

|----------------| "
@ 36,120 say &md
@ 36,120 say ln15 picture '99,999,999'
@ 36,120 say &ms
@ 37,1 say    "16. ITEM 5 MULTIPLIED BY ITEM 6b
                                                        |
| "
@ 37,103 say &md
@ 37,103 say ln16 picture '99,999,999'
@ 37,130 say &ms
@ 38,1 say    "17. LESSER OF ITEM 15 OR ITEM 16

|----------------| "
@ 38,120 say &md
@ 38,120 say ln17 picture '99,999,999'
@ 38,130 say &ms
@ 39,1 say     "18. TOTAL AMOUNT OF PREVIOUS PROGRESS PAYMENTS
REQUESTED
    | "
@ 39,120 say &md
@ 39,120 say ln18 picture '99,999,999'
@ 39,130 say &ms
@ 40,1 say    "19. MAXIMUM BALANCE ELIGIBLE FOR PROGRESS PAYMENTS
(Item 17 less 18)
| "
@ 40,120 say &md
@ 40,120 say ln19 picture '99,999,999'
@ 40,130 say &ms
@          4    1    ,    1          s    a    y
"-----------------------------------------------------------------
-----------------------------------------------------------------
----"
@ 42,1 say    "                                        SECTION III -
COMPUTATION OF LIMITS FOR OUTSTANDING PROGRESS PAYMENTS
```

```
                "
@ 43,1 say      "                                    * SEE SPECIAL
INSTRUCTIONS  ON  BACK  FOR  USE  UNDER  THE  FEDERAL  ACQUISITION
REGULATION.                    "
@           4    4    ,    1           s    a    y
"-----------------------------------------------------------------
-----------------------------------------------------------------
----"
@  45,1  say       "20.  COMPUTATION  OF  PROGRESS  PAYMENT  CLAUSE
(a(3)(i) or a(4)(i)) LIMITATION *                                |
        | "
@ 46,1 say     "     a. COSTS INCLUDED IN ITEM 11, APPLICABLE TO
DELIVERED, INVOICED, AND                                  | $
   | "
@ 46,100 say &md
@ 46,105 say ln20a picture '99,999,999'
@ 46,130 say &ms
@ 47,1 say     "        ACCEPTED TO THE DATE IN HEADING OF SECTION
II.                                                      |
 | "
@ 48,1 say      "      b. COSTS ELIGIBLE FOR PROGRESS PAYMENTS,
APPLICABLE TO UNDELIVERED ITEMS                              |
      | "
@ 49,1 say     "         AND TO DELIVERED ITEMS NOT INVOICED AND
ACCEPTED (Item 11 less 20a)                                |
    | "
@ 49,100 say &md
@ 49,105 say ln20b picture '99,999,999'
@ 49,130 say &ms
@ 50,1 say     "     c. ITEM 20b MULTIPLIED BY 6a

|----------------|$"
@ 50,120 say &md
@ 50,120 say ln20c picture '99,999,999'
@ 50,130 say &ms
@ 51,1 say     "     d. ELIGIBLE SUBCONTRACTORS PROGRESS PAYMENTS
(Item 14e)
  | "
@ 51,120 say &md
@ 51,120 say ln20d picture '99,999,999'
@ 51,130 say &ms
@ 52,1 say     "      e. LIMITATION a(3)(i) or a(4)(i) (Item 20c
plus 20d) *
    | "
@ 52,120 say &md
@ 52,120 say ln20e picture '99,999,999'
@ 52,130 say &ms
@  53,1  say       "21.  COMPUTATION  OF  PROGRESS  PAYMENT  CLAUSE
(a(3)(ii) or a(4)(ii)) LIMITATION *
|----------------| "
@ 54,1 say     "     a. CONTRACT PRICE OF ITEMS DELIVERED, ACCEPTED
AND INVOICED TO DATE IN                                  |
```

```
| "
@ 54,100 say &md
@ 54,105 say ln21a picture '99,999,999'
@ 54,130 say &ms
@ 55,1 say    "        HEADING OF SECTION II
                                                    |
| "
@ 56,1 say     "     b. CONTRACT PRICE OF ITEMS NOT DELIVERED,
ACCEPTED AND INVOICED (Item 5 less 21a)                  |
     | "
@ 56,100 say &md
@ 56,105 say ln21b picture '99,999,999'
@ 56,130 say &ms
@ 57,1 say    "     c. ITEM 21b MULTIPLIED BY ITEM 6b

|----------------| "
@ 57,120 say &md
@ 57,120 say ln21c picture '99,999,999'
@ 57,130 say &ms
@ 58,1 say    "     d. UNLIQUIDATED ADVANCE PAYMENTS PLUS ACCRUED
INTEREST
 | "
@ 58,120 say &md
@ 58,120 say ln21d picture '99,999,999'
@ 58,130 say &ms
@ 59,1 say    "     e. LIMITATION (a(3)(ii) or a(4)(ii)) (Item 21c
less 21d) *
| "
@ 59,120 say &md
@ 59,120 say ln21e picture '99,999,999'
@ 59,130 say &ms
@ 60,1 say    "22. MAXIMUM UNLIQUIDATED PROGRESS PAYMENTS (Lesser
of item 20e or 21e)
|----------------| "
@ 60,120 say &md
@ 60,120 say ln22 picture '99,999,999'
@ 60,130 say &ms
@ 61,1 say     "23. TOTAL AMOUNT APPLIED AND TO BE APPLIED TO
REDUCE PROGRESS PAYMENT                                  |
     | "
@ 61,100 say &md
@ 61,105 say ln23 picture '99,999,999'
@ 61,130 say &ms
@ 62,1 say    "24. UNLIQUIDATED PROGRESS PAYMENTS (Item 18 less
23)
|----------------| "
@ 62,120 say &md
@ 62,120 say ln24 picture '99,999,999'
@ 62,130 say &ms
@ 63,1 say    "25. MAXIMUM PERMISSIBLE PROGRESS PAYMENTS (Item 22
less 24)
| "
```

```
@ 63,120 say &md
if .not. mart
     @ 63,62 say   "First Article Limitation Applies"
     if ln25 > 0
          ln25 = min(mk1art,ln25)
     else
          ln25 = mk1art
     endif
     @ 63,120 say ln25 picture '99,999,999'
     ln26 = min(ln25,ln19)
else
@ 63,120 say ln25 picture '99,999,999'
endif
@ 63,130 say &ms
@ 64,1 say    "26. AMOUNT OF CURRENT INVOICE FOR PROGRESS PAYMENT
(Lesser of Item 25 or 19)
| "
@ 64,120 say &md
@ 64,120 say ln26 picture '99,999,999'
@ 64,130 say &ms
@ 65,1 say    "27. AMOUNT APPROVED BY CONTRACTING OFFICER

| "
@           6    6    ,    1           s    a    y
"-----------------------------------------------------------------
-----------------------------------------------------------------
----"
@ 67,1 say    "
           CERTIFICATION "
@ 69,1 say       "I certify that the above statement (with
attachments) has been prepared from the books and records of the
above-named contractor"
@ 70,1 say       "in accordance with the contract and the
instructions hereon,  and to the best of my knowlege and belief,
that it is correct, "
@ 71,1 say     "that all the costs of the contract performance
(except as herewith reported in writing) have been paid to the
extent shown herein, "
@ 72,1 say    "or where not shown as paid have been paid or will
be paid currently,   by the contractor,   when due,   in the
ordinary course of"
@ 73,1 say    "business,  that the work reflected above has been
performed,    that the quantities and amounts involved are
consistent with the"
@ 74,1 say     "requirements of the contract. That there are no
encumbrances (except as reported in writing herewith,   or on
previous progress"
@ 75,1 say    "payment request No.        ) against the property
acquired or produced for,  and allocated or properly chargeable
to the contract"
@ 76,1 say     "which would affect or impair the Government's
title,  that there has been no materially adverse change in the
```

```
financial condition"
@ 77,1 say    "of the contractor since the submission of the most
recent written information dated            by the contractor to
the Government"
@ 78,1 say     "in connection with the contract,  that to the
extent of any contract provision limiting progress payments
pending first article"
@ 79,1 say    "approval,  such has been complied with,  and that
after the making of the requested progress payments the
unliquidated progress"
@ 80,1 say    "payments will not exceed the maximum unliquidated
progress payments by the contract.  "
@          8    2    ,    1          s    a    y
"-------------------------------------------------------------
-------------------------------------------------------------
----"
@ 83,1 say       "NAME  AND  TITLE  OF  CONTRACTOR  REPRESENTATIVE
SIGNING THIS             |SIGNATURE "
@ 84,1 say    "FORM
                  |"

@ 85,1 say    "
                  |"
if mflag
     @ 85,25 say &md
     @ 85,83 say [DRAFT COPY ONLY, NOT FOR SUBMISSION !!!]
     @ 85,130 say &ms
endif
@ 86,1 say    "
                  |"
@          8    7    ,    1          s    a    y
"-------------------------------------------------------------
-------------------------------------------------------------
----"
@ 88,1 say    "NAME AND TITLE OF CONTRACTING OFFICER
                  |SIGNATURE "
@ 89,1 say    "
                  |"
@ 90,1 say    "
                  |"
@ 90,10 say &md
@ 90,10 say macotitl
@ 90,130 say &ms
@ 91,1 say    "
                  |"
@          9    2    ,    1          s    a    y
"-------------------------------------------------------------
-------------------------------------------------------------
----"
@ 93,1 say    "NSN 7540-01-140-5523
1443-101                                          STANDARD FORM
1443 (10-82)     "
```

```
@ 94,1 say     "
                                                    Prescribed  by
GSA(FPR 1-16.808) "
@ 95,1 say     "
                                                    FAR  (48  CFR
53.232)"
@ 96,1 say &mcr
@ 96,1 say &mr
@ 96,1 say &mff
*
set device to screen
*
do base
if mflag
     @ 3,30 say [SP-1443 Progress Payment]
     @ 10,12 say [A Draft P P R has been generated.]
     @ 11,12 say [If the request is acceptable, Enter a Y]
     @ 12,12 say [Enter N to start over ... ] get mans picture
"Y"
     read
     clear gets
     store .f. to mflag
     if .not. mans
          close databases
          release all like m*
          return
     endif
else
     @ 3,30 say [SP-1443 Progress Payment]
     @ 10,12 say [Updating Contract Files]
     store .f. to mflag
     store .f. to mans
endif
*
enddo
*
*
************************
select 4
if ln26> 0
     mpramt = ln26
else
     mpramt = ln19
endif
append blank
replace pcod with mpcod
replace prno with mprno
replace pprdt with mpprdt
replace pramt with mpramt
replace pdcst with mpdcst
replace pincst with mpcinst
replace pskppd with mpskppd
```

```
replace psklpd with mpsklpd
replace pskwait with mpskwait
replace pcntry with mpcntry
replace ptcst with mptcst
replace pciv with mpciv
replace pknum with contract
************************
select 2
replace ktcitd with ln12a
replace kpectd with ln9
replace kiectd with ln10
replace kskptd with ln14a
replace kskltd with ln14b
replace kpptd with mkpptd + mpramt
************************
release all like M*
close databases
sysr = "***"
sysk = "none selected"
do base
return

*eof ppr.prg
* isp.prg
* Author: Walt Harsch
*
*
*
*
*
*
*
clear
store space(17) to contract
store .t. to mansa
store .f. to mansb
store " " to choice
do base
@ 3,24  say [SP-1443 Invoice/Shipment/Payment]
@ 9,5   say [Enter Contract Number ... or Return to exit: ] get
contract picture '@R !!!!!!-!!-!-!!!! !!!!'
read
if contract = " "
     return
endif
@ 11,20 say [1    Record an Invoice]
@ 12,20 say [2    Record a Shipment]
@ 13,20 say [3    Record a Payment]
@ 14,20 say [9    Return to Main Menu]
@ 16,20 say [Enter your selection: ] get choice
read
sysk = contract
```

```
clear
do base
do case

************************
*  Invoice             *
************************

case choice = '1'
@ 3, 28 say [SP-1443 Invoice]
select 1
use ka index ka
restore from ka additive
seek contract
store kfms to kkfms
store 'US' to mkcntry
use inv index inv
restore from inv additive
do while .not. mansb
     @   7,12 say [       Enter Invoice Number: ] get minvono
picture '@R !!!-!!!!'
     @  8,12 say [       Enter Invoice Date: ] get minvodt
     @   9,12 say [       Enter Invoice Amount: ] get minvamt
picture '99999999.99'
     if kkfms
          @ 10,12 say [            Enter FMS code: ] get mkcntry
picture '!!'
     endif
     read
     @ 11,12 say [     Is All Data Correct?: ] get mansb picture
'Y'
     read
enddo
store .f. to mansb
select 2
use invr
do while mansa
     store space(7) to minvosh
     @ 13,12 say [         Shipment # Invoiced: ] get minvosh
picture '@R !!!-!!!!'
     read
     append blank
     replace iknum with contract
     replace invono with minvono
     replace invosh with minvosh
     @ 15,12 say [Are there more Shipments?: ] get mansa picture
'Y'
     read
     clear gets
enddo
select 1
append blank
```

```
replace iknum with contract
replace invono with minvono
replace invodt with minvodt
replace invamt with minvamt
release all like m*
close databases

************************
* Shipments            *
************************

case choice = '2'
@ 3,28  say [SP-1443 Shipment]
select 1
use ka index ka
restore from ka additive
seek contract
store kfms to mkfms
store "US" to mkcntry
use ship
restore from ship additive
do while .not. mansb
     @   5,12 say [        Enter Shipment Number: ] get mshipno
picture '@R !!!-!!!!'
     @  6,12 say [        Enter Shipment Date: ] get mshipdt
     read
     @   8,12 say [         Is All Data Correct ?: ] get mansb
picture 'Y'
     read
enddo
select 2
use shipr
store .t. to mansa
store .f. to mansb
do while mansa
     do while .not. mansb
          store space(7) to msclin
          mshipqty = 0
          @ 10,12 say [        Enter CLIN # Shipped: ] get msclin
picture '!!!!!!!'
          @  11,12  say  [Enter  CLIN  Quantity  Shipped:  ]  get
mshipqty picture '999999'
          read
     @ 13,12  say [          Is All Data Correct?: ] get mansb
picture 'Y'
          read
     enddo
     store .f. to mansb
     append blank
     replace sknum with contract
     replace shipno with mshipno
     replace sclin with msclin
```

```
     replace shipqty with mshipqty
     @ 15,12 say [         Are there more CLINs?: ] get mansa
picture 'Y'
     read
     clear gets
enddo
select 1
append blank
replace sknum with contract
replace shipno with mshipno
replace shipdt with mshipdt
release all like m*
close databases

************************
* Payments             *
************************

case choice = '3'
@  3,28 say [SP-1443 Payments]
select 1
use pay
restore from pay additive
store .f. to mansb
store .t. to mansa
do while .not. mansb
     @  8,12 say [    Enter Check Number: ] get mpyckno picture
'999999999999999'
     @  9,12 say [     Enter Check Date: ] get mpydate
     @ 10,12 say [    Enter Check amount: ] get mpyamt picture
'99999999.99'
     @ 11,12 say [Enter Discount Amount: ] get mpydisc picture
'99999999.99'
     read
     @ 13,12 say [Is All Data Correct?: ] get mansb picture 'Y'
     read
     append blank
     replace pyknum with contract
     replace pyckno with mpyckno
     replace pydate with mpydate
     replace pyamt with mpyamt
     replace pydisc with mpydisc
enddo
release all like m*
close databases

case choice = '9'
clear
do base
return

endcase
```

```
sysk = 'none selected'
clear
do base
return* mod.prg
* Author: Walt Harsch
*
*
*
*
do base
store .t. to mansa
store .f. to mansb
PUBLIC contract
store space(17) to contract
@ 3,27 say [SP-1443 Contract Modification]
@ 10,5 say [Enter Contract Number ... or Return to exit: ] get
contract picture "@R !!!!!!-!!-!-!!!! !!!!"
read
if contract = " "
     return
endif
sysk = contract
mmodamt = 0
do base
select 1
use mod index mod
restore from mod additive
select 2
use modr index modr
restore from modr additive
select 3
use clin index clin
restore from clin additive
set filter to cknum=contract
do while mansa
     do while .not. mansb
          @ 3,28  say [SP-1443 Modification Screen]
          @ 7,12   say [Modification Number: ] get mmodno picture
'!!!!!!!'
          @ 9,8    say [          CLIN #: ] get mmoclin picture
'!!!!!!!'
          @ 9,48 say [Date of Mod.: ] get mmoddt
          @ 10,8  say [Non-$ mods.?: ] get mmoadmin picture 'Y'
          read
          seek mmoclin
*         if eof()
*              @ 14,8 say [CLIN not found...Press any key]
*              minkey = 0
*              do while minkey = 0
*                   minkey = inkey()
*              enddo
```

```
*               clear
*               do base
*               loop
*          endif
           store cqtyc to mmoqtyf
           store cqtyc to mmoqtyt
           store cpricec to mmoprf
           store cpricec to mmoprt
           store cduec to mmoduef
           store cduec to mmoduet
           select 2
           @ 12,8  say [     Old Qty: ]
           @ 12,26 say mmoqtyf picture '999999'
           @  12,48  say [        New Qty: ] get mmoqtyt picture
'999999'
           @ 13,8  say [   Old Price: ]
           @ 13,26 say mmoprf picture '999,999.99'
           @  13,48  say [     New Price: ] get mmoprt picture
'999,999.99'
           @ 14,8  say [Old Due Date: ]
           @ 14,26 say mmoduef
           @ 14,48 say [New Due Date: ] get mmoduet
           read
           @ 16,22 say [Is All Data Correct?: ] get mansb picture
"Y"
           read
           mmodamt = mmodamt + ((mmoqtyt*mmoprt)-(mmoqtyf*mmoprf))

      enddo
      @ 16,22 say [Are there more CLINs to modify?: ] get mansa
picture "Y"
      read
      if .not. mansa
           select 1
           append blank
           replace modno with mmodno
           replace moddt with mmoddt
           replace moknum with contract
           replace modamt with mmodamt
           replace moadmin with mmoadmin
      endif
      select 2
      append blank
      replace modno with mmodno
      replace moknum with contract
      replace moclin with mmoclin
      replace moqtyf with mmoqtyf
      replace moqtyt with mmoqtyt
      replace moprf with mmoprf
      replace moprt with mmoprt
      replace moduef with mmoduef
      replace moduet with mmoduet
```

```
     select 3
     if eof()
          append blank
     endif
     replace cqtyc with mmoqtyt
     replace cpricec with mmoprt
     replace cduec with mmoduet
enddo
select 4
use kb
store knum to mknum
locate for knum = contract
do while knum = contract .and. .not. eof()
     store kcntry to mkcntry
     select 3
     mxtprice = 0
     store cknum to mcknum
     locate for cknum = contract .and. ccntry = mkcntry
     do while cknum = contract .and. ccntry = mkcntry .and. .not.
eof()
          store cqtyc to mcqtyc
          store cpricec to mcpricec
          store (mcqtyc*mcpricec)+mxtprice to mxtprice
          continue
     enddo
     select 4
     replace kpricec with mxtprice
     continue
enddo
release all like m*
close databases
do base
return

*eof mod.prg* recon.prg
* Author: Walt Harsch
* Purpose:  To reconcile a given contract
*
*
*
*
do base
store space(17) to contract
store .f. to mans
do while .not. mans
     @ 3,24 say [SP-1443 Contract Reconciliation]
     @ 10,9 say [Enter Contract Number (or CR to Exit): ] get
contract picture '@R !!!!!!-!!-!-!!!! !!!!'
     read
     if contract = " "
          loop
     endif
```

```
     use ka index ka
     restore from ka additive
     seek contract
     if .not. found()
          @ 12,12 say [Contract not found ... Press any key to
try again]
          minkey = 0
          do while minkey = 0
               minkey = inkey()
          enddo
          do base
          loop
     endif
store kgtotal to mdok
store kdate to mkdate
*
use kb index kb
seek contract+'US'
store kpplr to mkpplr
use recon
append from pay for pyknum = contract
append from ppr for pknum = contract
append from inv for iknum = contract
append from mod for moknum = contract
*
replace all rdate with invodt for invodt > ctod('01/01/01')
replace all rdate with pydate for pydate > ctod('01/01/01')
replace all rdate with moddt for moddt > ctod('01/01/01')
replace all rdate with pprdt for pprdt > ctod('01/01/01')
*
set index to rdate
reindex
*       Print Subroutine.
do base
@ 3,30 say [SP-1443 Recon Print]
@ 7,12 say [Ready Printer, press any key to print]
minkey = 0
do while minkey = 0
     minkey = inkey()
enddo
restore from epson additive
set device to print
@ 1,1 say &mr
@ 1,1 say &mc
mtpramt=0
mtmodamt=0
mtpyamt=0
mtpydisc=0
mtinvamt=0
mp = 0
munliq = 0
go top
```

```
do while .not. eof()
     ml = 7
     mp = mp + 1
     do while ml <= 63
          if ml = 63
               eject
               loop
          endif
          if ml = 7
               @ 1,1 say &mr
               @ 1,1 say &mc
               @   3,30 say [CONTRACT RECONCILIATION REPORT FOR
CONTRACT #: ]
               @  3,83 say contract picture '@R !!!!!!-!!-!-!!!!
!!!!'
               @  3,105 say date()
               @  3,120 say [Page: ]
               @  3,126 say mp picture '999'
               @  5,1   say [Date]
               @  5,10  say [Activity]
               @  5,23  say [$ Due on Cont.]
               @  5,38  say [Unliquidated $]
               @  5,53  say [Prog. Pay Req.]
               @  5,68  say [  +(-) K Mods.]
               @  5,83  say [ Payments Rec.]
               @  5,98  say [     Discounts]
               @  5,113 say [      Invoices]
               mi=1
               do while mi <= 130
                    @ 6,mi say "="
                    mi=mi+1
               enddo
          endif
          if ml=7 .and. mp=1
               @ ml,1  say mkdate
               @ ml,10 say [New Contract]
               @ ml,23 say mdok picture '99,999,999.99'
               ml=ml+1
               loop
          else
               @ ml,1 say rdate
               if pramt > 0
                    @ ml,10 say [PPR # ]
                    @ ml,16 say prno
                    @ ml,20 say pcntry
               endif
               if modamt > 0
                    @ ml,10 say [MOD # ]
                    @ ml,16 say modno
               endif
               if pyamt > 0
                    @ ml,10 say [CHK # ]
```

```
                    @ ml,08 say pyckno
               endif
               if invamt > 0
                    @ ml,10 say [INV # ]
                    @ ml,16 say invno
               endif
          endif
          mdok = mdok + modamt - pyamt
          munliq = max(munliq+pramt - ((mkpplr/100)*pyamt),0)
          @ ml,23 say mdok         picture '99,999,999.99'
          @ ml,38 say munliq   picture '99,999,999.99'
          @ ml,53 say pramt        picture '99,999,999.99'
          @ ml,68 say modamt  picture '99,999,999.99'
          @ ml,83 say pyamt        picture '99,999,999.99'
          @ ml,98 say pydisc  picture '99,999,999.99'
          @ ml,113 say invamt picture '99,999,999.99'
          mtpramt = mtpramt + pramt
          mtmodamt = mtmodamt + modamt
          mtpyamt = mtpyamt + pyamt
          mtpydisc = mtpydisc + pydisc
          mtinvamt = mtinvamt + invamt
          ml = ml + 1
          if .not. eof()
               skip
          else
               exit
          endif
     enddo
enddo
*
mi=1
do while mi <= 130
     @ ml,mi say [-]
     mi = mi + 1
enddo
ml = ml + 1
@ ml,1 say [TOTALS]
@ ml,53 say mtpramt   picture '99,999,999.99'
@ ml,68 say mtmodamt  picture '99,999,999.99'
@ ml,83 say mtpyamt   picture '99,999,999.99'
@ ml,98 say mtpydisc  picture '99,999,999.99'
@ ml,113 say mtinvamt picture '99,999,999.99'
eject
set device to screen
release all like m*
zap
close databases
return
*
* eof recon.prg
```

APPENDIX B

SP-1443 USER'S MANUAL

SP-1443

A Program for Managing Contract Progress Payments

USER'S MANUAL

This manual assumes that the user has a basic familiarity with the personal computer, its operating system (DOS) and the basic concept behind Government contract progress payments. If this is not the case, the computer user's manual, the DOS user's manual, The Federal Acquisition Regulation and Standard Form 1443 (progress payment form) should provide enough information to use the program effectively.

As with any new software product, it is wise to make a duplicate copy of the program disk prior to usage as a "backup." This program is free of any copy protection and can be duplicated without special hardware or software. The program can be run from either a hard disk or floppy drive.

ABOUT THE PROGRAM: Written in the winter of 1988 as part of a thesis on contract management and automation, this program sets up "files" on the disk that contain basic contract information. There is virtually no limit to the number of contracts or contract actions that can be recorded from the software point of view, but disk space could become

an issue. The user is cautioned to periodically check to insure that the data disk in use has a reasonable amount (10,000 bytes or so) of free space remaining. Additional data diskettes may be created by copying all of the non-program files (everything except SP-1443.EXE) from the distribution diskette to a blank, formatted diskette. Backups of data diskettes may be made by the normal copy or backup procedure.

This program will generate a hard copy form that very closely resembles a SF-1443 form, suitable for submission and payment. In addition, it will track such contract activity as Shipments, Invoices, Payments and Modifications. A second report, the Contract Reconciliation, is a chronological history of the program from the day it was signed to the date the report is generated. This report summarizes the financial activity of the contract, and shows such information as unliquidated progress payments, and total discounts taken, just to name two.

CONVENTIONS: In this manual, text presented inside the angle brackets, < >, is intended to be typed on the keyboard on the computer. In the same manner <return> indicates that a carriage return or "enter" should be typed.

SET UP PROCEDURES: To load the program, simply type <SP-1443> and <return>. The first screen you will see is the Main Menu (see Figure 6).

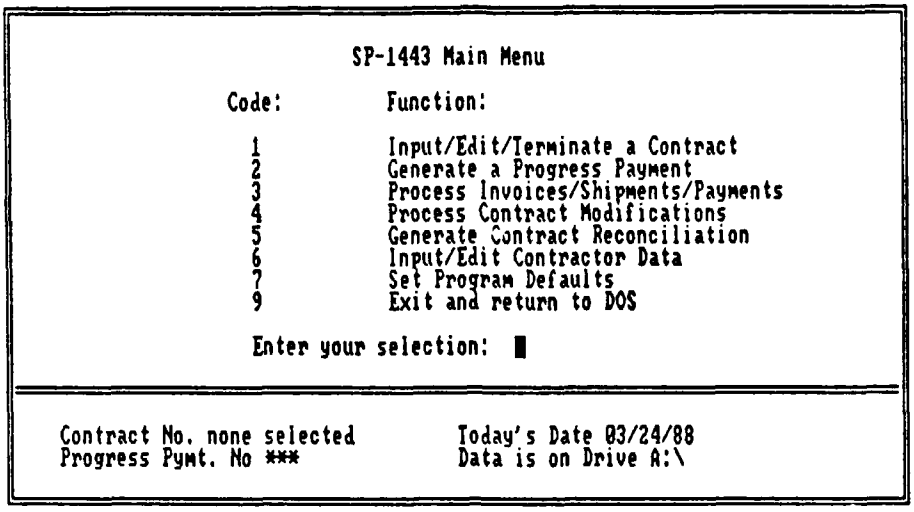


Figure 6. SP-1443 Main Menu

All of the screens in **SP-1443** will have the same general format. The screen will be titled at the top so the user has a fair idea of where he or she is in the program at any given time. The bottom status block will reflect pertinent program information, and will change as different contracts are selected etc.

From the Main Menu, select option 7, "Set Program Defaults." This will present another screen that will prompt you for the disk drive and path that you intend to use for your data. The program is initially set for drive A:\. It is important that you enter a valid drive and path, as the program will use what you type in literally (Figure 7). There is no need to do this every time you run the program, but it can be changed any time you desire.

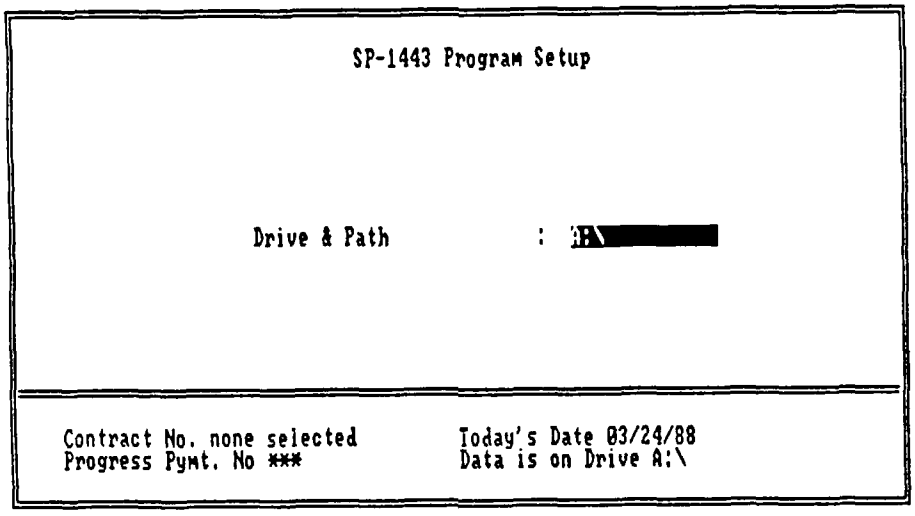


Figure 7. SP-1443 Program Setup

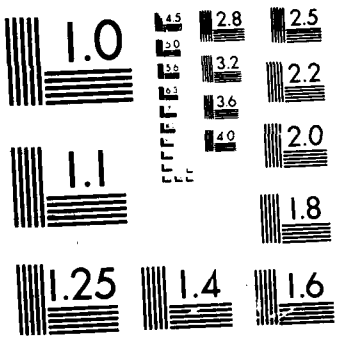
Remember to copy all files except the SP-1443.EXE file to the data drive or path that you select. These files contain the formats that will be needed to save your contract data. The program will add, modify or delete information in the files, but cannot create them from scratch.

The first time you run the program, it will automatically create some "*.MEM" files. These speed up operations for the program later. There is nothing the operator needs to do, it is a fully automatic operation.

The other initial selection that is supported on this screen is the printer selection. Currently Epson compatible dot matrix printers and Hewlett Packard compatible laser printers are supported. You may select either one prior to

AD-A194 661 AUTOMATED GOVERNMENT CONTRACT MANAGEMENT AS A PARADIGM FOR STANDARD PROGRAMS VERSUS STANDARD FORMS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA W W HARSCH MAR 88 2/2

UNCLASSIFIED F/G 5/1 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
10.0

printing, once a selection is set, it will stay set until you specifically change it again.

INPUTTING A CONTRACT: Option 1 will bring up a new screen that will ask you for a contract number. The system is set up to automatically convert lower case letters to upper case. Up to 17 characters (letters and or numbers) can be used. The standard DoD contract numbering scheme is portrayed for clarity, however, the data is stored as a single string of characters without hyphens or blanks. There is no need to insert them when entering a contract number. If you are entering a non-DoD contract number, simply type it in and press <return>. Similarly, contracts with delivery order numbers will use all 17 spaces, but those without will leave some blanks. Again, just enter the number and press <return>.

When the contract number is entered, another small menu will appear (Figure 8). Select the option you desire. **CAUTION**; to preserve data integrity, this system will let you edit a contract only once! If a typographical, or any other error is detected after the first edit, you must use main menu option #4, and insert an administrative contract modification. Any numbering convention will be accepted, but using **C00001** for the first "internal" contractor mod will make it easier to follow. Use the actual modification numbers for contract modifications signed by the government.

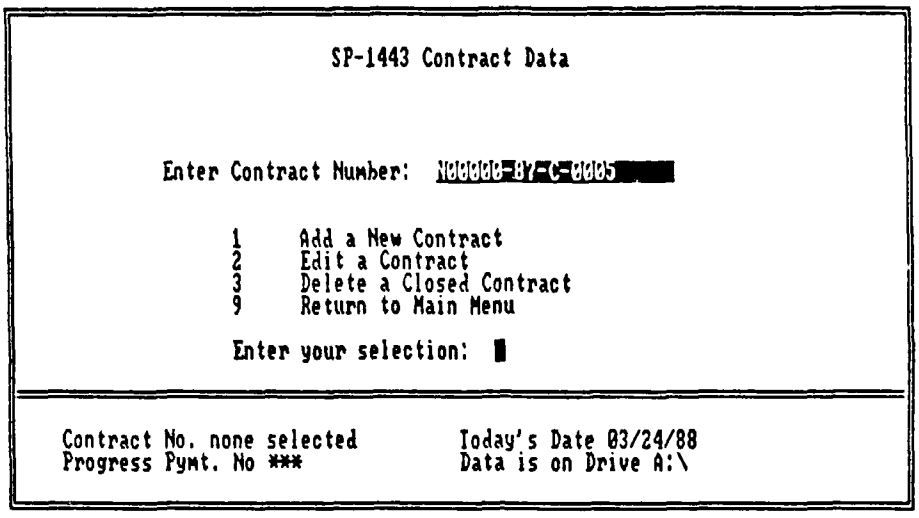


Figure 8. SP-1443 Contract Data

The next Contract Data Screen will ask for the Date of the contract and if there is FMS or Subcontracts contained in the contract (Figure 9). The next section asks for the First Article dollar limit (if applicable) and the Progress Payment and Liquidation Rates. These amounts are entered as whole numbers, i.e., 80% would be entered as <80> <return>.

The next two screens ask for ACO and Paying Office data (Figures 10 & 11). Most of this is self explanatory. The last two items however, I.D. Codes and Telex numbers may need further explanation. The "code" numbers can be found on the front page of your contract. If it is a Standard Form 26, look in the upper right hand corner of blocks 5 (or 6) 12. These codes uniquely identify each Government activity. The telex numbers refer to an "electronic mail

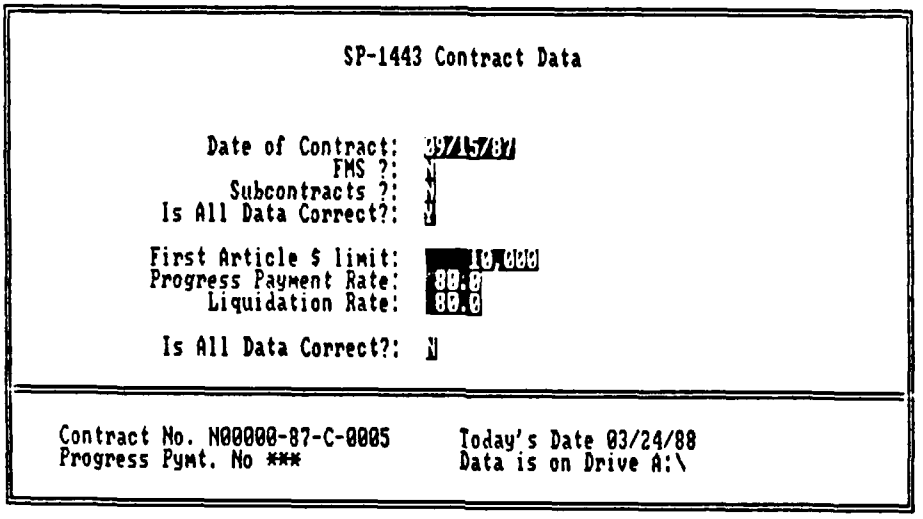


Figure 9. SP-1443 Contract Data

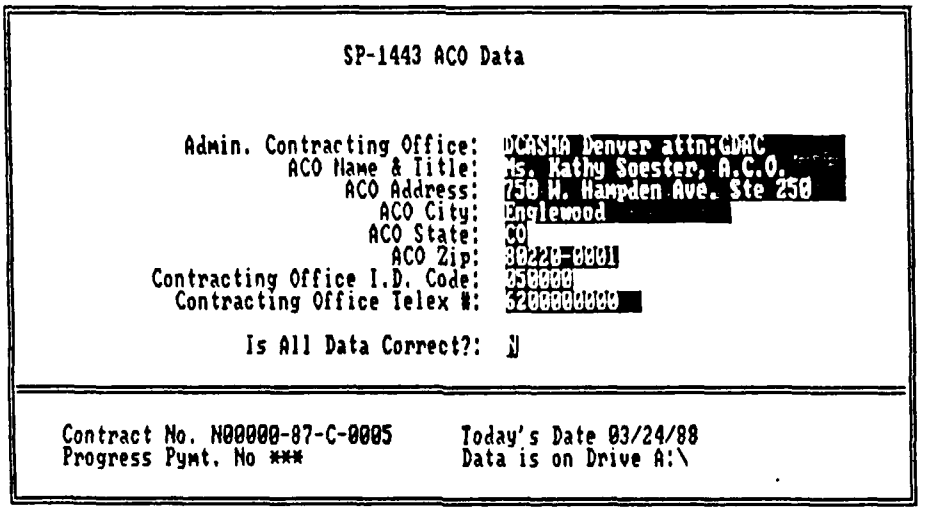


Figure 10. SP-1443 ACO Data

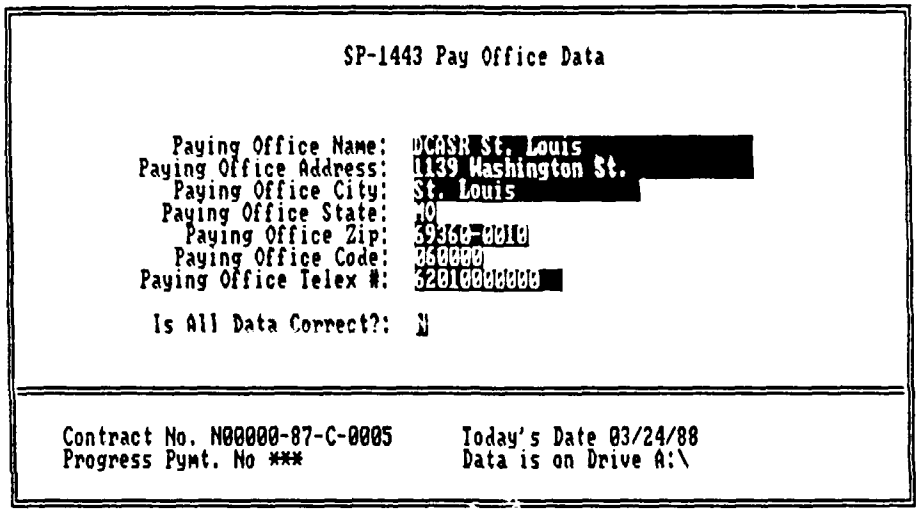


Figure 11. SP-1443 Pay Office Data

number" used to route commercial message traffic. Both of these entries are optional and will not effect the hard copy SF-1443.

You will note that the program will not ask you for the contract value. In order to maintain closer track on contract status, you will be asked to enter each of the CLIN's that have funds attached. The program will "roll up" the values, and when you answer the "Are there more CLINs?" question with <N>, the total contract price will appear on the screen. This figure should equal the face amount of the contract. If it does not, then double check the entries and edit the contract. You may enter CLINs without monetary value "not separately priced" if you wish, they will have no

effect on the program, but they will show up as "delivered" (if they have been) on the Contract Reconciliation form under option 5 (Figure 12).

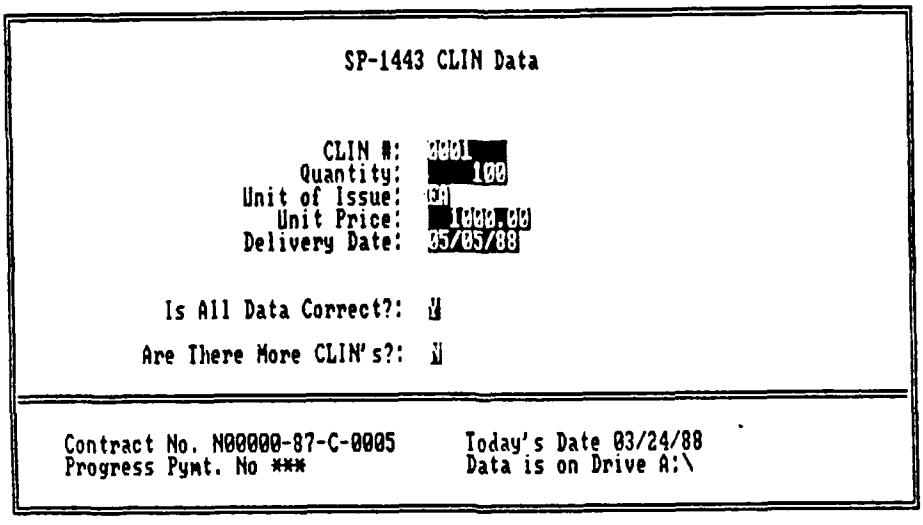


Figure 12. SP-1443 CLIN Data

INPUTTING CONTRACTOR DATA: This option is needed only once. It records your company's name, address etc., and infrequently changing data (whether or not you arc a Small Business, for example). This information can be edited at any time, but once entered, it is recorded on disk, and does not have to be re-entered unless something changes.

GENERATE A PROGRESS PAYMENT: This is the meat of the program. Considerable effort has gone into reducing the amount of user information needed to produce a complete and valid progress payment request. As with most main menu

options, the user is asked to enter a contract number. If the contract you select has FMS (foreign military sales) on it, then a country code will be requested. The country code is a two character code that should be associated with the various CLINs (Contract Line Item Numbers) in your contract. Because the Government must keep the "books" for these countries (and later bill them), this program treats each country as though it had a separate contract with you. Each country will start at 001 for the progress payment associated with it. Frankly, FMS can get a little complicated. If you have questions, please contact your Administrative Contracting Officer (ACO). This program was designed to keep these accounts separate for accounting purposes, and allows different payment and liquidation rates for each country. If there is FMS on a contract, the country code will appear next to the progress payment request number in block 8a of the form. In an effort to simplify the progress payment request process, the amounts that are asked for on the progress payment request screen (Figure 13) are for the "current period" only. That is to say, only the incremental costs (paid, incurred, etc.) that have accumulated since the last progress payment request need to be entered. Since the program keeps track of the running totals, there is no need to compute any "to date" amounts. If there are no subcontractors involved, the

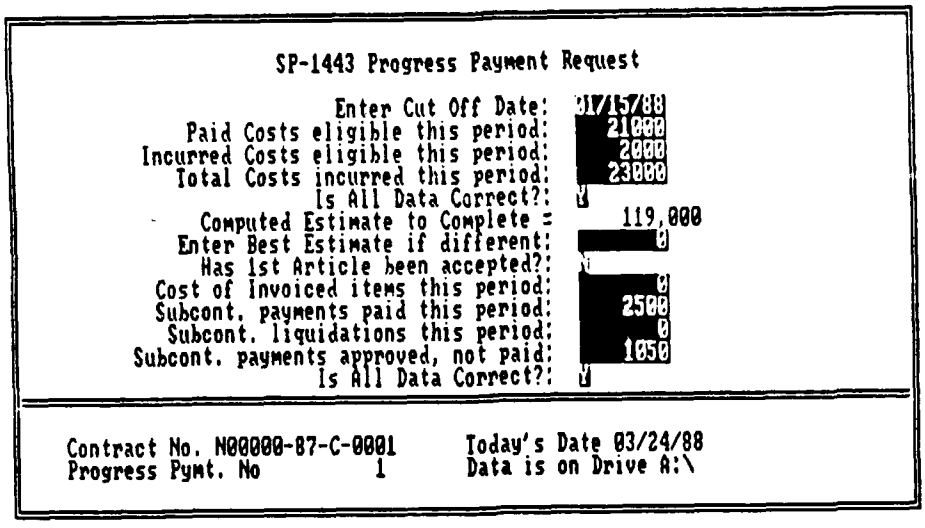


Figure 13. SP-1443 Progress Payment Request

subcontract related questions will not be asked, and zeros entered in the appropriate blocks of the final form.

IMPORTANT: After the current period costs have been collected, the program will compute the balance of funds remaining on the contract and display that amount on the screen. If your best estimate to complete the contract is different, enter that amount in the corresponding block. **BE ADVISED:** if you enter a larger amount, you are effectively saying that the contract may be in a loss position. The program will detect this and ask you if there are any unpriced or undefinitized contract actions, or if there are pending but not yet received contract mods. These amounts will be entered automatically into the formula prescribed in

the Federal Acquisition Regulation (FAR) part 35 regarding loss position contracts. If the contract is still in a loss position, the loss ratio will be automatically computed, and the figures on the resulting progress payment request will reflect that ratio.

Conversely, if you feel that a significantly smaller estimate to complete is appropriate, the program will take no action other that reporting it on the form. The ACO should be apprised of the situation however, so that funds can be deobligated and released for other possible procurements.

Most of the numbers are self explanatory on this screen, however two bear explanation. 1. **TOTAL COSTS INCURRED THIS PERIOD:** refers to the "grand total" for the period. At a minimum it will be equal to the sum of the two previous amounts (incurred and paid costs). It could however be greater. There may be costs that are not allowable or allocable under the contract, but are experienced none the less. This figure is used to arrive at the estimate to complete mentioned above, so give this area some thought. 2. **COST OF INVOICED ITEMS:** This is not the "face value" of current invoices, it is your "cost" (presumably a smaller figure). This will have to come from your accounting records, it cannot be derived from existing data in the program.

As a control function, a "draft copy" of the progress payment request will be printed with a message appearing in the signature block stating that it is for review only and not for submission. This is a "forced" function, you cannot get the final copy without going through the "draft" phase. Use it to double check for mistakes. The whole point of the program in the first place is to reduce the number of forms returned for error correction.

INVOICES, SHIPMENTS AND PAYMENTS: As with previous screens, a contract number will be requested. If a corresponding contract is found, you will be asked which type of transaction you wish to enter. Figures 14, 15 and 16 show the input screens. There is no forced format for

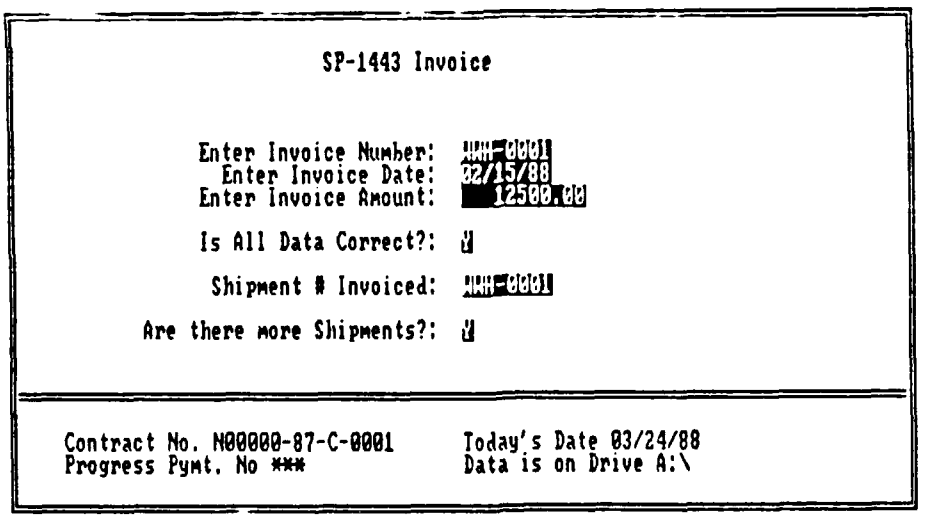


Figure 14. SP-1443 Invoice

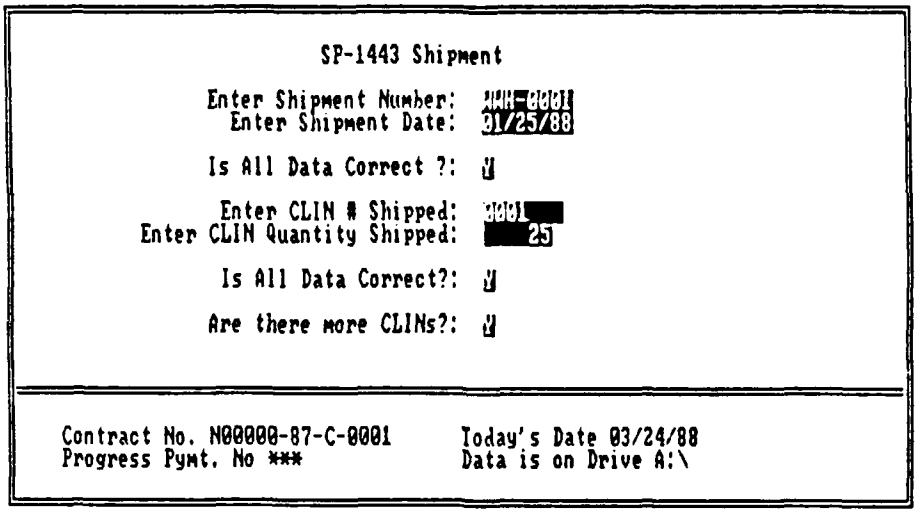


Figure 15. SP-1443 Shipment

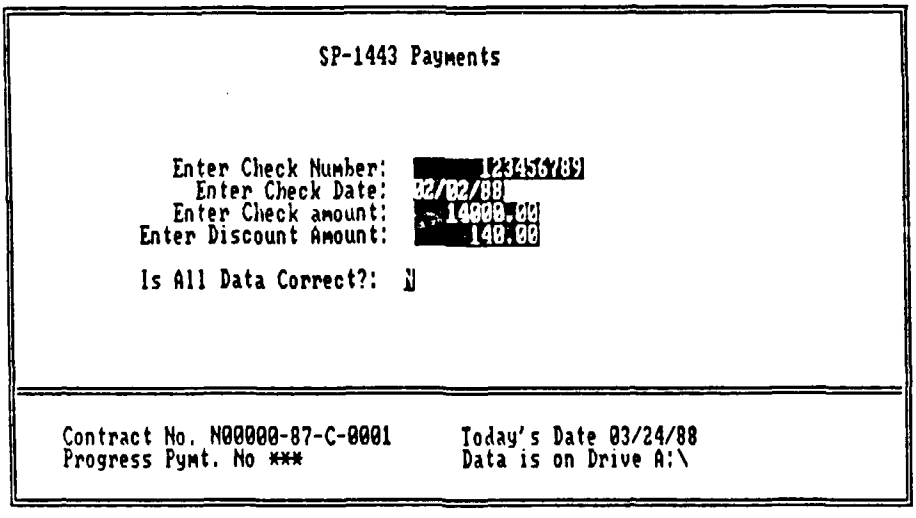


Figure 16. SP-1443 Payments

invoice or shipment numbers. The hyphen supports the convention of three letters and four digits. For example, General Motors fourth invoice might be **GMC-0004**. The computer will accept any seven characters in these fields.

CONTRACT MODIFICATIONS: Once the contract has been selected, the contract mod screen is selected (Figure 17). As with the original contract, monetary values are determined at the CLIN level. The current values for price, quantity and due date are displayed on the left side of the screen. Only those items you actually change will be changed on the disk (skip the items that remain the same by pressing <return>). Some modifications may not effect money. Skip the CLIN block, and answer yes <Y> to the no $ change question. This will allow the user to make one more edit session on the contract section.

GENERATE CONTRACT RECONCILIATION: As discussed in the beginning of this manual, the reconciliation option lets the user create a "spreadsheet" like chronological summary of the contract. All that is required of the user is to input a contract number and the program does the rest. This function is totally automated, and there are not user inputs required.

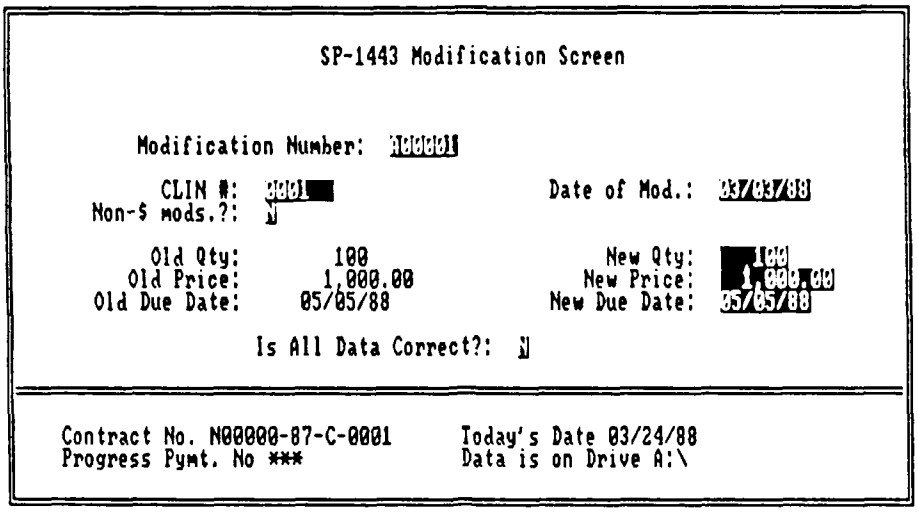


Figure 17. SP-1443 Modification Screen

LIST OF REFERENCES

1. U.S. Naval Audit Service, Audit Report #A20076, NAVPRO Bethpage, New York, 30 June 1987.

2. U.S. Department of Defense, Inspector General, Report on the Audit of the DoD Policies on the Use of Progress Payments, Report No. 85-086.

3. U.S., Federal Acquisition Regulation, Part 32, Section 503-3.

4. Cash, James I. Jr., and others, Corporate Information Systems Management: Text and Cases, 2d Ed., p. 403, Irwin, 1988.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information System Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Director, Information Systems (OP-945) Office of the Chief of Naval Operations Navy Department Washington, D.C. 20350-2000	1
4.	Computer Technology Programs, Code 37 Naval Postgraduate School Monterey, California 93943-5000	1
5.	Professor Daniel R. Dolk, Code 54Dk Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1
6.	Professor Y. Ben Mortagy, Code 54My Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1
7.	LCDR R.W. Smith, Code 54Sx Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1
8.	Professor T.R. Sivasankaran, Code 54Sj Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1
9.	LT Walter W. Harsch, SC, USN c/o Naval Audit Service P.O. Box 1206 Falls Church, Virginia 22041-0204	12

END

DATE

FILMED

8-88

DTIC