```
0010 REM %
*** P R O G R A M M I N G   T O O L S ***
```

SYSTEM 2200

PMi.06-2200.02A-00FI-1-1

SYSTEM 2200 PROGRAMMING TOOLS

June 17, 1974 — Revision 3

This revision contains both an improved general utility package, and a revised compression program.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

NOTE:

To obtain a program tape of the referenced subroutines in both the source and compressed versions, send your order for "System 2200 Programming Tools" - Tape No. 701-0128B to:

WANG Laboratories, Inc.
836 North Street
Tewksbury, Mass. 01876
ATTN: General Services Dept.

The price of the tape is $20.00, or $25.00 for both the General Utilities Manual and the tape.

TABLE OF CONTENTS

		PAGE
CHAPTER I	GENERAL INFORMATION	
1.1	INTRODUCTION	1-1
1.2	CLASSES OF UTILITIES/SUBROUTINES	1-1
	Class 1 - Stand-alone Utilities	1-1
	Class 2 - Resident Utilities	1-1
	Class 3 - Subroutines	1-2
CHAPTER II	DEFINITIONS: OVERVIEW OF THE APPLICATION ARCHITECTURE	
2.1	INTRODUCTION	2-1
2.2	SYSTEM	2-1
2.3	PROGRAMS	2-1
2.4	MODULE	2-1
2.5	FILE	2-1
2.6	REPORTS	2-2
CHAPTER III	NAMES	
3.1	INTRODUCTION	3-1
3.2	PROGRAM NAME	3-1
	First Three Characters	3-1
	Fourth Character	3-1
	Fifth and Sixth Characters	3-1
	Seventh and Eighth Characters	3-2
3.3	FILE NAME	3-2
3.4	VARIABLES	3-3
CHAPTER IV	PROGRAM STRUCTURE	
4.1	INTRODUCTION	4-1
4.2	IDENTIFICATION	4-1
4.3	FORMAT	4-4
4.4	SAMPLE PROGRAM	4-5
4.5	SOURCE PROGRAM COMPRESSION	4-8
CHAPTER V	FILE STRUCTURE	
5.1	INTRODUCTION	5-1
5.2	IDENTIFICATION CODES	5-3
5.3	HEADER AND TRAILER RECORDS	5-5
	Application Header Record	5-5
	Warning Record	5-5
	End-of-Volume/End-of-File Record	5-5
	Header/Trailer Conventions	5-7
	Writing a File	5-7
	Reading a File	5-7

TABLE OF CONTENTS

		PAGE
5.4	DATA RECORD DESIGN TECHNIQUES	5-8
	Size of Logical Records	5-8
	Record Type Indicator	5-8
	System 2200 Record Description Form	5-10
5.5	KEY FIELDS	5-11
5.6	LONG RECORDS	5-14
5.7	SAMPLE CODING	5-14
	Initialize Cassette	5-15
	DEFFN' 240 Open Output File	5-15
	DEFFN' 241 End-of-Volume/File Output	5-15
	DEFFN' 250 Open File Input	5-16
	DEFFN' 251 Close Volume Input	5-16
CHAPTER VI	SCREEN USAGE TECHNIQUES	
6.1	INTRODUCTION	6-1
6.2	SCREEN LAYOUT	6-1
	Input Phase	6-1
	Execution Phase	6-2
	Output Phase	6-2
6.3	SCREEN USAGE UTILITIES	6-3
	Position Cursor	6-3
	Accept CRT Input	6-3
	Operator Wait	6-3
CHAPTER VII	REPORT PRODUCTION CONVENTIONS	
7.1	PRE-PRINTED CONTINUOUS FORMS	7-1
7.2	STANDARD, NON PRE-PRINTED, FLAT PACK FORMS (66 LINES/PAGE, 132 CHARACTERS/LINE)	7-2
CHAPTER VIII	DOCUMENTATION STANDARDS	
8.1	INTRODUCTION	8-1
8.2	SYSTEM LEVEL DOCUMENTATION	8-1
	Management Summary	8-1
	System Flowchart	8-1
	Machine Requirements	8-1
	Operating Instructions	8-1
	Estimation of Throughput	8-2
	Input Layouts	8-2
	Report Layouts	8-2
	File Layout	8-2
8.3	PROGRAM LEVEL DOCUMENTATION	8-2
	Narrative Description of Program	8-2
	Special Explanatory Information	8-3
	Logic Flowcharts	8-3
	Operating Instructions	8-3
	Document and File Layouts	8-3
	Source Program Listing	8-3
8.4	DOCUMENTATION FORMS	8-3

LIST OF ILLUSTRATIONS

Figure 2-1.	Program "System" for One Application	2-3
Figure 4-1.	Identification for a Segmented Module	4-3
Figure 5-1.	File Arrangement on Tape Blocks	5-4
Figure 5-2.	Header and Trailer Records	5-5
Figure 5-3.	System 2200 Record Description	5-9
Figure 5-4.	Key Fields of Various Record Types	5-11
Figure 5-5.	New England Power Company File	5-13
Figure 5-6.	Key Fields in Subrecords	5-14
Figure 5-7.	Writing Record	5-17
Figure 5-8.	Reading Record	5-18
Figure 7-1.	Alignment Marks	7-1
Figure 7-2.	Sample Print Mask	7-1
Figure 7-3	Normal Report Page Form	7-3

LIST OF TABLES

Table 3-1.	Program Name PAY1101 Defined	3-2
Table 3-2.	File Name PAY1F010 Defined	3-2
Table 4-1.	Identification Section of a Program	4-2
Table 5-1.	Record Types	5-6
Table 5-2.	Record Description Form Categories Defined	5-10
Table 6-1.	First Four Lines of CRT During Input Phase	6-2

LIST OF CHARTS

Chart 5-1.	New England Power Street Light Inventory	5-2

CHAPTER I

GENERAL INFORMATION

1.1 INTRODUCTION

Programming tools are provided as an aid in the production of orderly programs. The conventions are intended to aid the programmer from the conception of the problem to the actual publication of the manuals. For example, with the guidelines:

the structure and physical appearance of programs is consistent from programmer to programmer, allowing subsequent parties to follow the logic of all program systems more easily;

adopted names for programs, data and files have a rational meaning, facilitating finding, using and, if necessary, modifying these items;

screen usage and dialogue syntax is consistent from program to program and from application to application, allowing the operating instructions to be assimilated more easily by the end user.

In order to make the System 2200 as easy to use as possible, a group of utility programs and subroutines have been developed, such as the BASIC EDIT PROGRAM, a cursor positioning utility, et cetera. Use of the utilities and subroutines (described in System 2200 General Utilities) eliminates repetitive coding and encourages observance of the guidelines outlined in this manual.

1.2 CLASSES OF UTILITIES/SUBROUTINES

There are three classes of utilities/subroutines.

Class 1 Stand-alone Utilities

Stand-alone utilities are programs, as the name implies, entirely and solely resident within the System 2200; an example is a compression program which is loaded into the machine and allows a program tape to be reduced in size and recorded in the condensed version on a second tape. While the compression utility is being run, the compression program is the only program resident within the System 2200.

Class 2 Resident Utilities

Resident Utilities always stay in memory when an application program is run. They include such routines as open file, close file, position cursor, et cetera.

Class 3 Subroutines

Subroutines are a group of continually evolving routines to aid programmers in developing applications. Subroutines remain resident within the System 2200 only during the time they are being employed; that is, they are called from the tape or disk when necessary. A frequently used subroutine in the business world is conversion of the Gregorian date to a Julian date and vice versa.

CHAPTER II

DEFINITIONS: OVERVIEW OF THE APPLICATION ARCHITECTURE

2.1 INTRODUCTION

Historically, applications for a calculator were written as stand-alone, one-shot programs. It was not necessary for other subsequent programs to use data generated by previous programs. Futhermore, no one ever envisioned a "system" of programs for one application (see Figure 2-1).

With the advent of the System 2200 and its improved data handling capability, more complicated problems can be solved, often requiring several programs or systems of programs. Because of this added complexity, a group of common definitions have been adopted. The tools used by all programmers must be the same to produce a uniform product.

2.2 SYSTEM

The system consists of all programs, files, reports, et cetera needed for a given application. For example, payroll, invoicing, and inventory control are separate systems within the general framework called accounting.

2.3 PROGRAMS

A given system such as payroll requires many programs. For example, in a large payroll system, the data entry is a separate operation from printing of checks. It is conceivable, even likely, that data entry and check printing may be separate programs, run at different times and on separate program tapes.

2.4 MODULE

A module is a program overlay. Because of core limitation, an entire program may not be kept resident within the System 2200. It is usually necessary to call upon additional core loads of instructions from the program tape. Each core load or overlay is called a module. A typical module is a routine to calculate federal withholding tax.

2.5 FILE

A file is a collection of related records treated as a unit. The data within the file can be read and updated by a program. For instance, a payroll file has the data for all employees. A record consists of all the data for a particular employee. Within each record are fields. Fields are the individual items contained within each record. A payroll record has a field for the name of the employee, the social security number, et cetera. To recapitulate, the fields make up the record, the records make up the file.

2.6 REPORTS

Reports are a means of presenting the results of a program; i.e., the checks written by the check writing program are an example of the many reports generated by a typical payroll system.

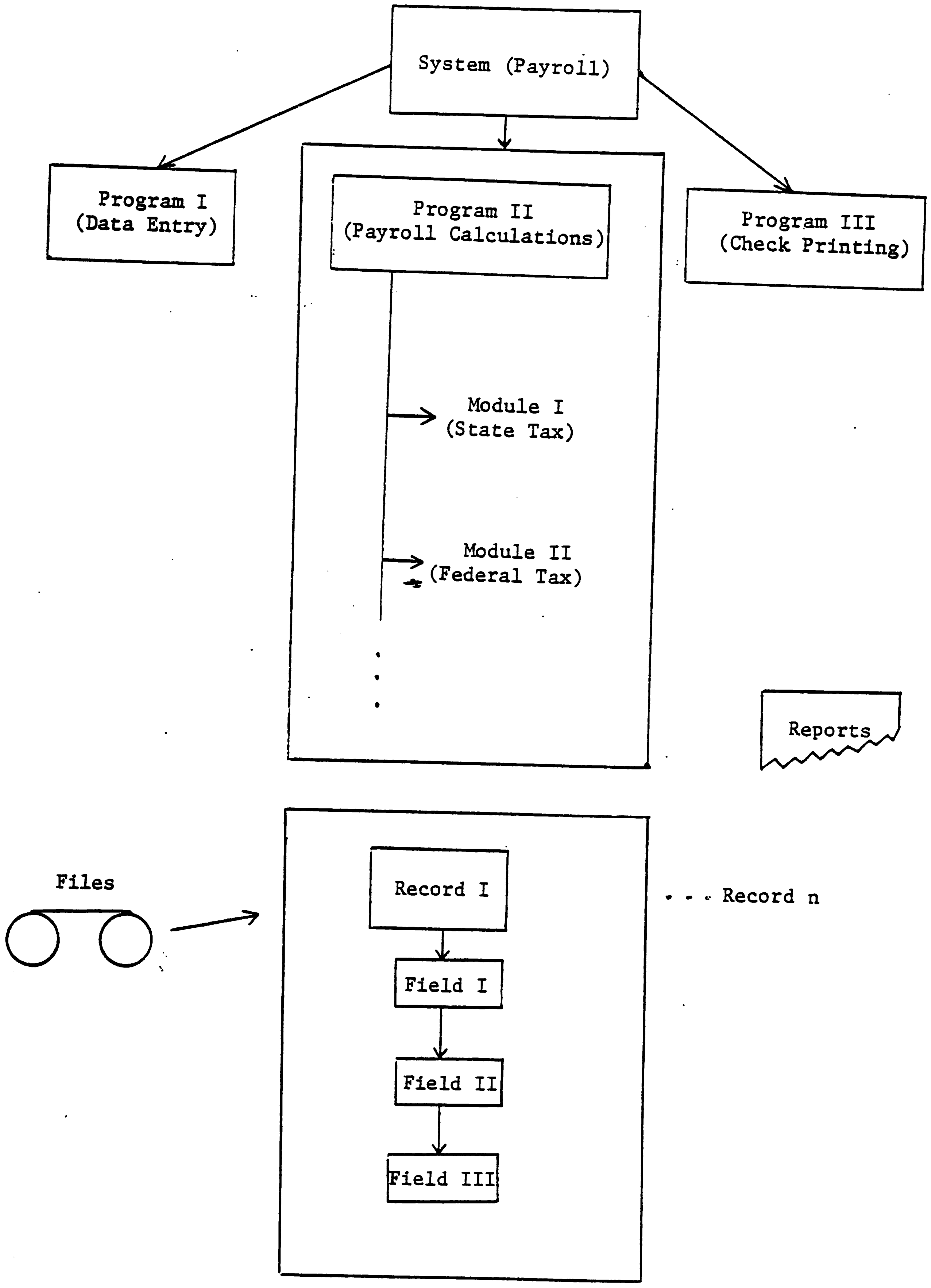


Figure 2-1. Program "System" for One Application

(This page intentionally left blank)

CHAPTER III

NAMES

3.1 INTRODUCTION

According to Webster, a name is a word constituting the distinctive designation of a person or thing. With this definition in mind, programs, modules, files and variables are named to make them easily identifiable and thus increase their usefulness.

3.2 PROGRAM NAME

To make a program name as meaningful as possible, programs are named in a consistent fashion. The program name must be eight characters long; each character represents a specific meaning (see Table 3-1).

First Three Characters

The first three positions are the system name in letters. Make these letters correspond as much as possible to the system name; e.g., Payroll Accounting = PAY.

Fourth Character

The next position (No.4) is a single digit indicating the target configuration of equipment on which the system is run. Three initial system configurations are described below.

1. System 2200A with 8K memory and three Tape Cassette Drives and a 132(or more) column output device. System No. 1 is the minimum configuration being proposed for any standard applications package. No. 1 allows one to copy files, update files, copy tapes, and sort data files. The output device can be either a line printer, high-speed printer, or an output writer.

2. System 2200B with 8K memory, three flexible disk drives and 132 (or more) column output device. The three flexible disk drives provide the same capabilities as No. 1, but eliminate the manual handling of tapes and provide larger volumes of rapidly accessible storage.

3. System 2200B with 8K memory, a fixed/removable disk drive and a 132 column (or more) output device. The fixed/removable disk provides larger volume storage with randomly accessible files.

Fifth and Sixth Characters

The next two positions (No. 5 and No. 6) are the program numbers within the system. Each system generally has more than one program.

Seventh and Eighth Characters

The next two positions (No. 7 and No. 8) are the module or overlay number within a program.

Table 3-1. Program Name PAY11011 Defined

Character Positions	Description	Example
1-2-3	System Name (in letters)	PAY
4	Equipment Configuration (one digit)	1
5-6	Program Number within System (two digits)	10
7-8	Module within Program (two digit code)	11

3.3 FILE NAME

In order to relate a file to a system and program, files also are named in a consistent fashion (see Table 3-2). All eight available character positions must be used. The first four positions are the same characters used in the program name, to denote the system and equipment configuration. The fifth character is the letter "F", indicating a file. The last three characters indicate numeric sequencing. Initially, sequencing is in steps of ten to allow for later insertion of new files.

Table 3-2. File Name PAY1F010 Defined

Character Positions	Description	Example
1-2-3	System Name (in letters)	PAY
4	Equipment Configuration	1
5	File Indicator - always letter "F"	F
6-7-8	Numeric Sequencing of Files - starting with 010 in steps of ten	010

3.4 VARIABLES

The WANG System 2200 BASIC variable check list and listing forms are intended to help maintain control over the variables used in a program.

As a variable is used, it must be checked off on the Check-Off List. To distinguish between different usages of variables, the following scheme must be employed (put the appropriate number in the cross-referenced box representing that variable name):

0 = non-common,
1 = declared common in this module,
2 = declared common in a previous module.

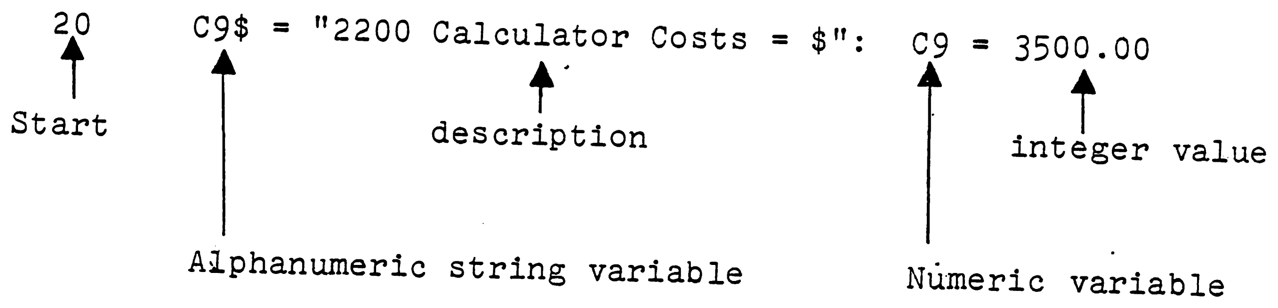
The listing sheet is used to write a verbal description of what each variable represents.

Always assign a unique variable for each unique item. With 1,144 variables to choose from, one should rarely run out of new "names".

A meaningful and illustrative series of numbers and letters should be used to help identify the character, thing, expression, et cetera which is being operated upon.

Use an alphanumeric string variable and a numeric variable similar in appearance for related items.

Example:



```
20      C9$ = "2200 Calculator Costs = $":  C9 = 3500.00
```

Thus the following PRINT statement is easier to interpret:

```
30 PRINT C9$; C9
```

Specific variables (all "Q's" and "W's") are reserved for system utilities. The programmer can reference, change or use the information contained in the variables as needed in his program, but must not change the meaning of the variables. For example, when using STANDARDS UTILITIES (see System 2200 General Utilities manual) Q6 represents ROW; one can add, subtract, multiply or divide the value of Q6, but Q6 must always represent the ROW position.

SYSTEM 2200 VARIABLE CHECK-OFF LIST

PROGRAM NAME ______________________ DATE ______________

VERSION ______________________ PROGRAMMER ______________

SYSTEM ____

NUMERIC SCALARS
FORMAT = MN

N \ M	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Δ																										
1																										
2																										
3																										
4																										
5																										
6																										
7																										
8																										
9																										
0																										

NUMERIC ARRAYS
FORMAT = MN(

N \ M	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Δ																										
1																										
2																										
3																										
4																										
5																										
6																										
7																										
8																										
9																										
0																										

ALPHA NUMERIC SCALARS
FORMAT = MN$

N \ M	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Δ																										
1																										
2																										
3																										
4																										
5																										
6																										
7																										
8																										
9																										
0																										

ALPHA NUMERIC ARRAYS
FORMAT = MN$(

N \ M	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Δ																										
1																										
2																										
3																										
4																										
5																										
6																										
7																										
8																										
9																										
0																										

NOTE:
0 = NON COMMON
1 = COMMON DEFINED BY THIS MODULE
2 = COMMON DEFINED BY PREVIOUS MODULE

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL.(617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3175
9-73-2.5C
Price $1.50

SYSTEM 2200 ALPHA-NUMERIC SCALARS

PROGRAM NAME ____________________ VERSION ____________

SYSTEM ____________________ DATE ____________

PROGRAMMER ________________________________

VARIABLE	DESCRIPTION

WANG LABORATORIES, INC.
836 NORTH STREET TEWKSBURY, MASSACHUSETTS 01876, TEL (617) 851-4111, TWX 710 343 6769 TELEX 94 7421

Printed in U.S.A.
700-3179
9-73-2.5C
Price $1.50

SYSTEM 2200 NUMERIC SCALARS

PROGRAM NAME ______________________ VERSION ______________

SYSTEM ______________________ DATE ______________

PROGRAMMER ______________________

VARIABLE	DESCRIPTION

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3178
9-73-2.5C
Price $1.50

SYSTEM 2200 ALPHA-NUMERIC ARRAYS

PROGRAM NAME ________________ VERSION ________

SYSTEM ________________ DATE ________

PROGRAMMER ________________

VARIABLE	DESCRIPTION

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY MASSACHUSETTS 01876 TEL (617) 851-4111, TWX 710 343 6769 TELEX 94 7421

Printed in U.S.A.
700-3176
9-73-2.5C
Price $1.50

SYSTEM 2200 NUMERIC ARRAYS

PROGRAM NAME ______________________ VERSION ______________

SYSTEM ______________________ DATE ______________

PROGRAMMER ______________________

VARIABLE	DESCRIPTION

WANG LABORATORIES, INC
836 NORTH STREET TEWKSBURY MASSACHUSETTS 01876, TEL (617) 851-4111, TWX 710 343 6769 TELEX 94 7421

Printed in U.S.A.
700-3177
9-73-2.5C
Price $1.50

CHAPTER IV

PROGRAM STRUCTURE

4.1 INTRODUCTION

A well-structured program, logically constructed, indented for legibility, commented wherever appropriate by the use of REM statements, is easier to debug, easier for the user to understand, and easier for someone other than the author to modify. The programmer generally has access to a System 2200 with a memory large enough to run a well-documented program; customers, in general, do not have such a large machine.

With the use of the BASIC EDIT PROGRAM (see System 2200 General Utilities) all extra spaces, all REM statements other than the first one in the program, and most unnecessary line numbers can be optionally removed. Unnecessary line numbers are removed by creating multiple statement lines. Thus, two tapes are supplied to a customer; the source tape and an execution tape (compressed).

Programmers tend to create the compressed version first. Invariably, the source program never materializes. As a result, the programs are incomprehensible to other programmers, and, after six months, even the author does not remember the program in detail. Source programs are necessary and must be written first.

4.2 IDENTIFICATION

Table 4-1 is an example of the identification section of a program. The information in the first statements of a program module must follow the required format.

Notice each statement begins with a REM. The first REM statement at the beginning of every module is the eight character program name and the version (revision) associated with that module. The REM statement remains even after running the BASIC EDIT PROGRAM. The next three REM statements expand the meaning of the module name.

The next two REM statements are the copyright information, followed by REM statements containing the original date when written and all revision dates. When the program is published and released to the field, a WANG library number is assigned. The library number must be in the next REM statement. The remaining REM statements (as many as needed), are used to fully expound and explain the program's operation, including files used and functions performed. See Table 4-1 for the proper format of identification information.

Table 4-1. Identification Section of a Program

FORMAT	EXAMPLE
10 REM PROGRAM NAME, VERSION = ___	10 REM PROGRAM NAME = PAY11011, VERSION = 03
20 REM SYSTEM NAME = ___	20 REM SYSTEM NAME = PAYROLL ACCOUNTING
30 REM PROGRAM FUNCTION = ___	30 REM PROGRAM FUNCTION = CALCULATE
40 REM MODULE NAME = ___	40 REM MODULE NAME = FEDERAL TAX
50 REM THIS PROGRAM IS A PART OF A GENERALIZED APPLICATION	
60 REM SYSTEM COPYRIGHT, WANG LABS, INC. 1975	
70 REM DATE WRITTEN = MM/YY	70 REM DATE WRITTEN = 06/73
80 REM REVISION DATE = MM/YY	80 REM REVISION DATE = 07/74
90 REM REVISION DATE = MM/YY	90 REM REVISION DATE = 07/75
100 REM CURRENT WANG LIBRARY # = ___	100 REM CURRENT WANG LIBRARY # = 1234567
110 REM THIS PROGRAM DOES...	110 REM THIS PROGRAM CALCULATES PAYROLL
120 REM A. FILES USED = ___	120 REM A. FILES USED = 1.EMPLOYEE LIST 2.TAX TABLE
130 REM B. FUNCTIONS PERFORMED = ___	130 REM B. FUNCTIONS PERFORMED = PRINTS CHECKS, ETC.

NOTE:

The identification must precede eve
(overlay) of a program.

A source program module can be so hig
cannot fit into the System 2200 without breaki
segments, even though the compressed program m
System 2200 intact (see Figure 4-1). In this
segment of the module need contain the identifi

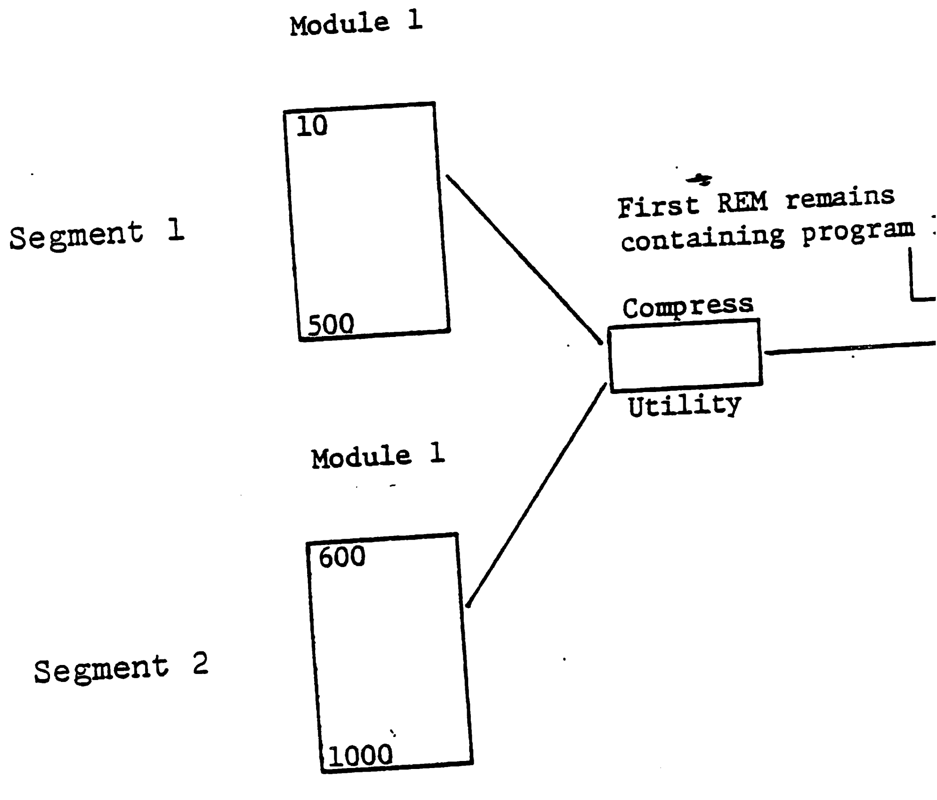


Figure 4-1. Identification for a S

When saving a program on tape, alway
name of the program contained in the first
overlays that are not new modules, as in Figu
tape by keying SAVE without the module number

4.3 FORMAT

In order to increase legibility, all statements in a program must be presented in a logical manner: only one statement to a line; indent to increase legibility, especially when performing FOR NEXT loops.

Example:

```
10      FOR I = 1 TO 10                    }
20          PRINT I                        }
30      FOR J = 1 TO 50                    } correct
40      PRINT I * J                        }
50      NEXT J                             }
60      NEXT I                             }

10      FOR I = 1 TO 10: FOR J = 1 TO 50  incorrect
```

Use REM statements liberally to comment the program.

Example:

```
10    REM   ***********************************
20    REM   * THE FOLLOWING SECTION COMPUTES  *
30    REM   * MONTHLY MORTGAGE PAYMENTS       *
40    REM   * INPUTS  N1 = NO. OF MOS.        *
50    REM   *         I1 = INTEREST RATE      *
60    REM   * OUTPUT  M1 = MORTGAGE PAYMENT   *
70    REM   ***********************************


90    REM        Calculating Payment
100                 M1 = N1 * I1
```

NOTE:

A REM statement always stands alone on a line; no other instructions are permitted on that line.

In general, a program consists of an initialization segment, a main segment and a wrap-up segment, in addition to one or more processing or input/output subroutines. By breaking a program into segments, labeling them carefully, and labeling the transfer points to the subroutines carefully, the legibility of the program is greatly enhanced.

Example:

```
190   REM   CALCULATE STATE TAX
200         GOSUB 675
210   REM   CALCULATE FEDERAL TAX
220         GOSUB 800
```

Following is a BASIC coding form, with a few lines filled in to illustrate its use. The tag field and the destination tag field are used to label transfer points mnemonically, rather than by statement number alone.

4.4 SAMPLE PROGRAM

The sample program provided below illustrates the format explained in Section 4.3.

```
 10    REM    PROGRAM NAME = BAN10111, VERSION = 10
 20    REM    SYSTEM NAME = BANK ACCOUNTING
 30    REM    PROGRAM FUNCTION = MORTGAGE
 40    REM    MODULE NAME = MONTHLY MORTGAGE PAYMENT COMPUTATION
 50    REM    THIS PROGRAM IS PART OF A GENERALIZED APPLICATION
 60    REM    SYSTEM COPYRIGHT, WANG LABS, INC. 1973
 70    REM    DATE WRITTEN = 06/70
 80    REM    DATE REVISED = 06/73
 90    REM    CURRENT WANG LIBRARY NUMBER = 987654
100    REM    THIS PROGRAM CALCULATES THE MONTHLY PAYMENT
116    REM    FOR A $40,000.00 MORTGAGE
120    REM    WHEN INTEREST VARIES FROM 7-1/2% to 9% IN 1/2%
              INCREMENTS
130    REM    AND THE NUMBER OF YEARS VARIES
140    REM    FROM 20 TO 30 YEARS IN 5 YEAR INCREMENTS
150           PRINT "AMOUNT BORROWED", "INTEREST RATE",
              "NO. OF YEARS", "MO. PYMT"
160    REM    INTEREST RATE VARIES BY 1/2%
170           FOR I = .075 TO .090 STEP .005
180    REM    YEARS OF REPAYMENT VARIES OVER EACH INTEREST RATE
190                 FOR N = 20 TO 30 STEP 5
200    REM    THE FOLLOWING CALCULATES AND PRINTS
210                     M = (40000*I/12)/(1-(1 + I/12)↑(N * 12))
220                     PRINT "$40000", 100 * I; "%", N, "$"; M
230                 NEXT N
240           NEXT I
250           STOP
260    REM$
```

(This page intentionally left blank)

SYSTEM NAME ____________________ DATE ____________________

PROGRAM NAME ____________________ VERSION ____________________

TAG	STMT NO.	BASIC STAT
BEGIN	10	GOSUB 300
	20	
	30	
	40	
	50	
	60	
RE-DO	70	GO TO 10
	80	
	90	REM THIS SECTION CALCULATES
	300	X = 5
	10	
	20	
	30	
	40	
	50	RETURN
	60	
	70	
	80	
	90	
	00	
	10	
	20	
	30	
	40	
	50	
	60	
	70	
	80	
	90	
	00	

SYSTEM 2200 CODING FORM

PAGE____ OF____

PROGRAMMER

IT	DESTINATION TAG
	TAX
	BEGIN
X E S	

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876 . TEL (617) 851-4111, TWX 710 343 6769 TELEX 94 7421

Printed in U.S.A.
700-3180A
1-74-1M
Price $2.50

4.5 SOURCE PROGRAM COMPRESSION

The "compressing" utility (BASIC EDIT PROGRAM) shrinks the size of a program (see System 2200 General Utilities for a detailed description). The three options available (one may employ any or all of them) are:

1. Remove unnecessary spaces.
2. Remove all REM statements except the first.
3. Combine several statements on one line.

In order to use BASIC EDIT, the program to be compressed must be in the proper format. The last statement in the module (which can be more than one segment) must be REM$. All program segments are compressed to form a single program.

After compression using all three options, the mortgage payment example (Section 4.4) reads:

```
10REMPROGRAMNAME=BAN10111,VERSION=10150PRINT "AMOUNT BORROWED",
"INTEREST RATE","NO. OF YEARS","MO PMT":FOR I=.075TO.090STEP .0
05:FOR N=20TO30STEP 5:M=(40000*I/12)/(1-(1+I/12)↑(-N*12)):PRINT
 "$40000",100*I;"%",N,"$";M:NEXT N:NEXT I:STOP
```

CHAPTER V

FILE STRUCTURE

5.1 INTRODUCTION

A file is an ordered list of information; a collection of related records treated as a unit. The structure of a file should be organized so the program/programmer can intelligently use this ordered list of information. The easiest way to discuss file structure is with an example, an electric company's inventory of street lights.

Typically, firms such as New England Power consist of several companies (Boston Edison, Maine Electric Power, et cetera). Each company is, in turn, broken into districts resembling the way a metropolitan area (Boston) phone book is written: Metro, North, West and South. Each district encompasses several towns. In Boston, Metro includes Cambridge, Arlington, Boston, Orient Heights. The towns are further subdivided by streets. Then, of course, each street has the ultimate listing of street lights, of which there may be several types, each having a different cost, replacement date, et cetera. Chart 5-1 illustrates the ordered concept of a file.

Indenting on the chart shows the hierarchical structure of information within a file. All similar items are indented the same amount. A street light on Inman Street in Cambridge has the same relative file position as a street light on Center Street in Newton.

The information from the chart can be directly mapped over to a tape file. Each line is a record. The items contained on the line are the fields of that record. The fields of the street light record are: type of light, date installed, cost of installation, et cetera.

Each indentation on the chart represents a different record type. Different record types are needed because the fields generally vary. The information contained under "I. Boston Edison Company" does not necessarily have the same format as the information of "a. Harvard Avenue".

In the chart, the five different record types are:

```
I.   Company Name
     A.   Division
          1.   Town or City
               a.   Street
                    i.     street light
```

The System 2200 is not capable of indenting, as one can on paper. Each type of record must be uniquely identified by means of a code stored within the record. The identification code is the first field of every record. Company record types (Boston Edison) would have a record identifier such as "Al", while street names, "Dl".

Chart 5-1. NEW ENGLAND POWER STREET LIGHT INVENTORY

```
I.    Boston Edison Company
      A.    Metro Boston District
            1.    Brookline
                  a.    Harvard Avenue
                        i(1).  Street light no. 1, Class a, etc.
                        i(2).  Street light no. 2, Class f, etc.
                         .
                         .
                         .
                        i(n).  Street light no. n, Class j, etc.
                  b.    Marion Street
            2.    Cambridge
                  a.    Inman Street
                        i(1).  Street light no. 1, Class y, etc.
                         .
                         .
                         .
                        i(n).  Street light no. n, Class p, etc.
                  b.    Massachusetts Avenue
                        i(1).
                         .
                         .
                         .
                        i(n).
      B.    West Suburban
            1.    Newton
                  a.    Center Street
                        i(1).
                         .
                         .
                         .
                        i(n).
                  b.    Washington Street
                        i(1).
                         .
                         .
                         .
                        i(n).
            2.    Wellesley
                  a.    Linden Street
                  b.    Rockland Street
      C.    Main Central Power
            etc.
```

Normally, the headings or major divisions always are known but the amount of records within each heading are not known in advance. One knows there are streets in a city, but not how many. After the last valid item of a particular record type is read, the System 2200 must be alerted the next item is a different record type. After all the lights on a street are listed, the system must be alerted for another street, or a warning record for End-Of-Volume (EOV), or End-of-File (EOF).

5.2 IDENTIFICATION CODES

The first character of the identification code is the record type (A = Company, D = Street Names). The second character of the ID code indicates, via a number 1, that it is a valid data record; a number 2, that it is the last data record of its type; and a number 3, that the next record is an EOV/EOF record.

Example:

A1	Boston Edison
A2	Last Company record in utility
A3	EOV/EOF is next

To further illustrate, consider the New England Power System with one company - Boston Edison; one division - Metro; one city - Cambridge; two streets - Inman and Mass Ave.; and three lights on each street - i(1), i(2), i(3). Assume the record types are labeled as follows:

A1	Company
A2	Division
C1	City/Town
D1	Street
E1	Light
E2	End of lights on street
D2	End of streets in city
C2	End of cities in division
B2	End of divisions in company
A2	End of companies in utility system

The file is written on the blocks of the cassette tape as illustrated in Figure 5-1.

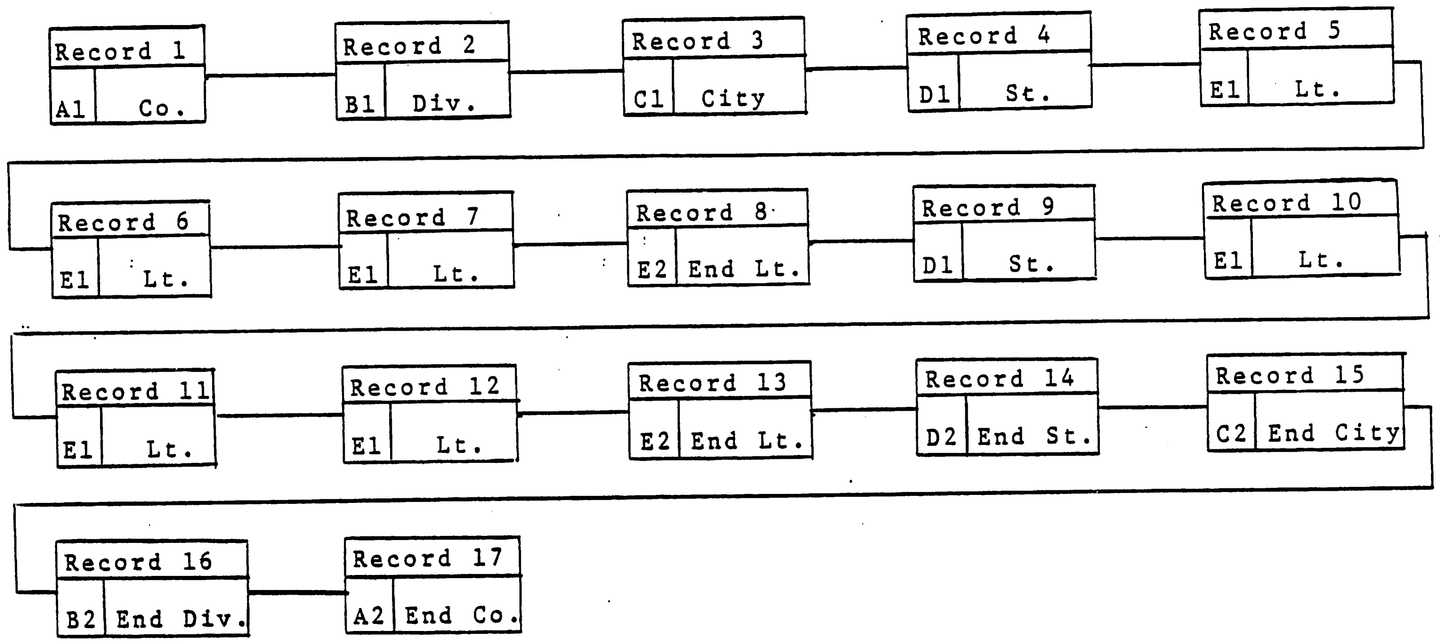


Figure 5-1. File Arrangement on Tape Blocks

The identification code ending with a "2" indicates the record is the last data record of a type. Type "2" records are written in the same format as Type "1" data records. The fields in the Type "2" records are checksum totals of the previous group of Type "1" records. For the System 2200A, only the numeric fields of the Type "2" records can contain the checksum indicator. The B version can checksum the alpha fields as well, using the ADD instruction. The checksum total is a software check on the hardware to ensure all tape reading and writing has been performed correctly.

5.3 HEADER AND TRAILER RECORDS

At the beginning and end of every file must be some special file header and trailer information. Header and trailer information is contained in separate records from the data records outlined above (see Figure 5-2.

Application Header Record

An application Header Record is generated by the program/programmer and contains information such as the date created, retention cycle of the file (how many days it must be kept before the tape can be reused), et cetera.

Warning Record

The Warning Record is written in the same format as the next expected data record, and tells the program the preceding data record was the last valid data record on the cassette. It also alerts the program to expect another record indicating whether more data for the file is on another volume, or if it is the end of the file.

When listing street lights, it is discovered there is not sufficient room on the volume to complete the file. A warning record is written, in the form of the light records, which alerts the program to expect a new kind of record. That new record, in this case, is an End-of-Volume record, telling the operator to change tapes in order to continue processing the data file.

End-of-Volume/End-of-File Record

The End-of-Volumel/End-of-File Record is a special block generated by a system utility to indicate the end of a volume or the end of a file.

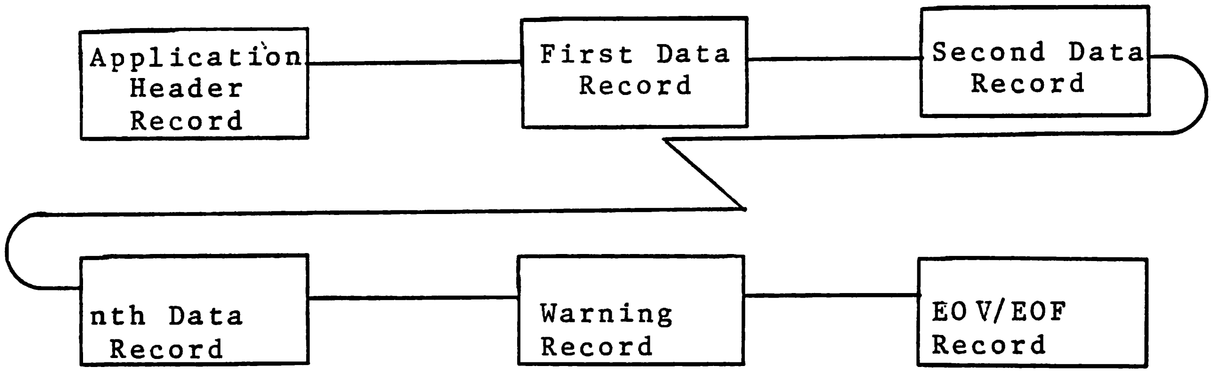


Figure 5-2. Header and Trailer Records

In the electric company example, ten record types were defined. Actually, seventeen record types are available (see Table 5-1).

Table 5-1. Record Types

Record Identifier	Description
--	Application Header Record
A1	Company
B1	Division
C1	City/Town
D1	Street
E1	Light
E2	End of Lights for Street
D2	End of Streets for City
C2	End of Cities for Division
B2	End of Divisions for Company
A2	End of Companies for System
E3	Warning Record in Light Format
D3	Warning Record in Street Format
C3	Warning Record in City Format
B3	Warning Record in Div. Format
A3	Warning Record in Co. Format
--	End-of-Volume/End-of-File Record

NOTE:

A1, A2 and A3 records must be in the same format. The same rule applies for B, C, et cetera, type records.

The software always compares the second position of the record ID type to see what type of record is coming: (1) valid data record; (2) end of record type - look for next logical record type to follow; and (3) a warning record saying an EOV/EOF is coming.

Example:

E1	is a valid street light record
E2	end of street light records
E3	is a warning that the next data record is an EOV/EOF record

NOTE:

All Type "2" records should contain the checksum totals for the fields of the preceding group of Type "1" records.

Header/Trailer Conventions

The DATASAVE OPEN instruction creates a special header record. DATASAVE END creates a special trailer record. These records are not sufficient to control multi-reel files, nor do they record creation dates or other pertinent information. As explained in the introduction, other records are necessary; therefore, DATASAVE OPEN and DATASAVE END are not used.

Writing a File

In place of the DATASAVE OPEN and DATASAVE END commands, two utilities should be used to generate all the necessary hardware and software header and trailer records, when writing a file.

1. Open Output File

2. End-of-Volume/File Output

These subroutines are explained in detail in the System 2200 General Utilities manual (STANDARDS UTILITIES).

Reading a File

When reading a file, two utilities are used:

1. Open File Input(to read the header records)

2. Close Volume Input (to indicate to the user whether the file is an End-of-File, or End-of-Volume, in which case, another volume must be loaded).

These also are explained in the General Utilities manual.

5.4 DATA RECORD DESIGN TECHNIQUES

Size of Logical Records

Each physical data record has a maximum of 253 bytes. Each field within a data record uses the following number of bytes:

a. length plus 1 for alpha fields ($)

b. 9 bytes for numeric fields (n)

To compute the volume in bytes for a given matrix, take the number of elements in that matrix and multiply by a. or b. above as appropriate.

Example 1:

An alpha array dimensioned DIM A$ (2,10) 5 has 20 elements of length 5, so its length is (2*10)*(5+1) = 120 bytes on the tape.

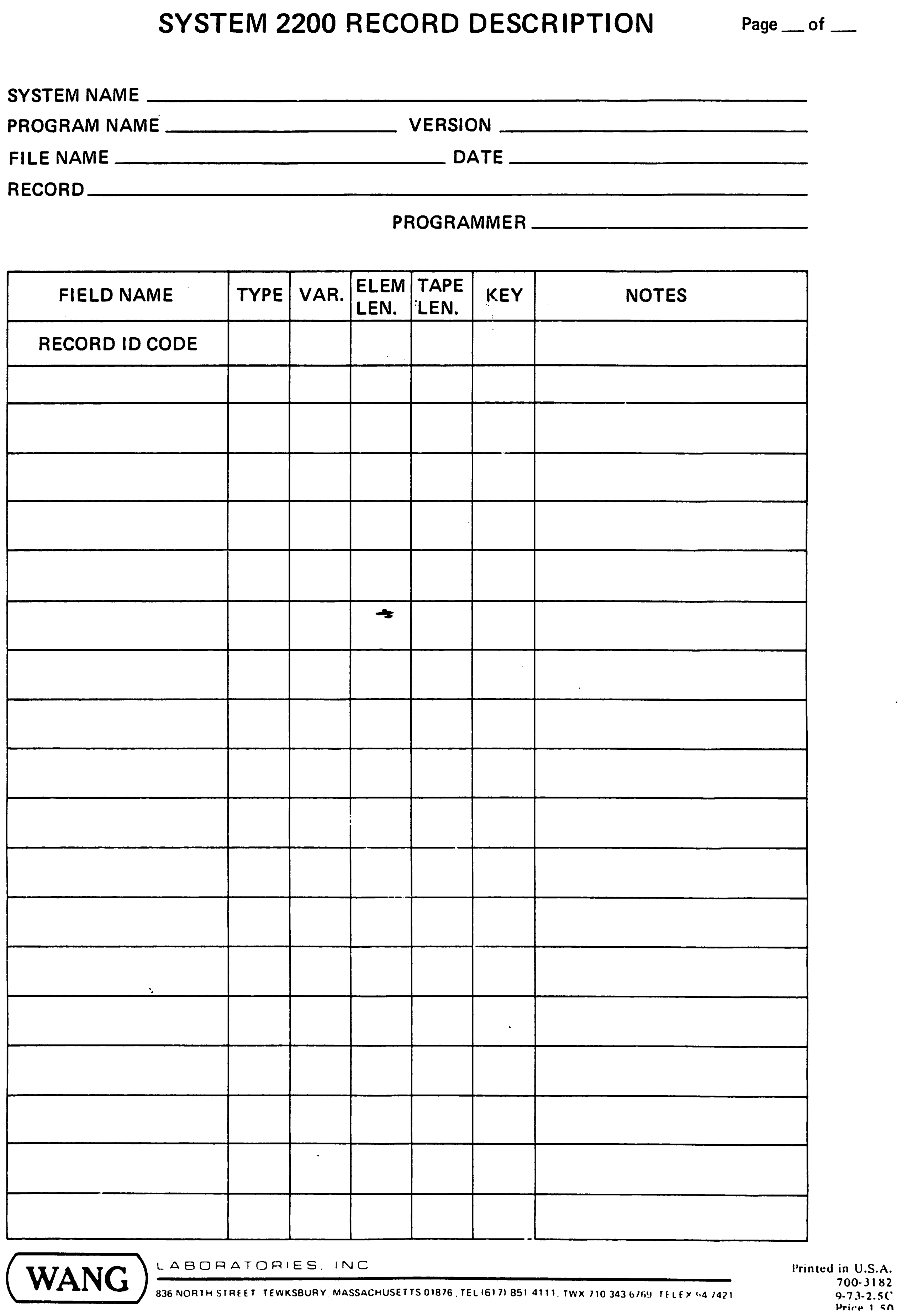
Example 2:

A similar numeric array dimensioned DIM A(2,10) has a length of (2*10) * 9 = 180 bytes on the tape.

Each data record should be approximately 240 bytes to allow for expansion without having to restructure the entire file. If records are considerably shorter than 240 bytes, they should be packed to total 240 bytes in order to maximize tape storage utilization. If the records are longer than 240 bytes, they should be broken into smaller records, with like fields grouped together in the same portion of the field.

Record Type Indicator

Each record must contain a two-byte record type indicator. The record type indicator supplies three kinds of information to the applications program: (1) what type of record follows; (2) when a particular type of record ends; and (3) a warning record for End-of-File or End-of-Volume. The record type indicator always is the first field in a record.



SYSTEM 2200 RECORD DESCRIPTION

Page __ of __

SYSTEM NAME ________________________________

PROGRAM NAME ________________ VERSION ________________

FILE NAME ________________ DATE ________________

RECORD ________________________________

PROGRAMMER ________________

FIELD NAME	TYPE	VAR.	ELEM LEN.	TAPE LEN.	KEY	NOTES
RECORD ID CODE						

WANG LABORATORIES, INC

836 NORTH STREET TEWKSBURY MASSACHUSETTS 01876, TEL (617) 851 4111, TWX 710 343 6769 TELEX 94 7421

Printed in U.S.A.
700-3182
9-73-2.5C
Price 1.50

Figure 5-3. System 2200 Record Description

System 2200 Record Description Form

The System 2200 Record Description form (Figure 5-3), a necessary part of application documentation, must be filled out for each type of record on a file. See Table 5-2 for an explanation of the form.

Table 5-2. Record Description Form Categories Defined

CATEGORY	DEFINITION
Application Name	same name as application program
Application ID	first 4 characters of program name, e.g., PAY1
File Name	8 characters, detailed in Section 3.3 (FILE NAME)
Date	when record designed
Record Name	short English description
Record ID Code	indicates record type, as in power company example, Boston Edison was A1
Field Name	English description of the field
Type	alpha or numeric field
Variable	name of variable (i.e., A, B, C$); for arrays, each variable within the array must be specified (e.g., B$ (3,4), G(5))
Element Length	length of the element; in the case of arrays, the length of the element within the array
Tape Length	for numerics, always 9 bytes; for alphas, always the length of the field plus one
Key	place a check (✓) if to be used as a key field
Notes	special remarks such as parameters of the field, how it relates to other fields, or purpose of the fields

5.5 KEY FIELDS

Within a record are one or more fields called key fields. Key fields are alpha or numeric data which uniquely identify the record. They can be used later as a basis for sorts (arrange the record in either ascending or descending alphabetic or numeric order based upon the key field). Examples of key fields are employee numbers, inventory parts numbers, et cetera.

Do not confuse the key fields with the two-byte record type which serves solely to indicate a particular kind of record within a file.

The key field(s) always are the first field(s) within a record and are arranged from major to minor immediately following the record-type code. As in the case of hierarchical files, all key fields should be reproduced from the highest level to the lowest level. Consider three record types labeled A1, B1, C1. Record Type A1 has a key field 01, Record Type B1 has key field 02 and Record Type C1 has key field 03 (see Figure 5-4).

Record Type	Keys			Other Fields	
A1	Key 1	LV	LV	Other Fields	Highest Member
B1	Key 1	Key 2	LV	Other Fields	
C1	Key 1	Key 2	Key 3	Other Fields	Lowest Member
C2	Key 1	Key 2	HV	Checksum	End of Record Type, Format same as C1
B2	Key 1	HV	HV	Checksum	End of Record, Format of B1
A2	HV	HV	HV	Checksum	End of Record, Format of A1

LV = Lowest Value
HV = Highest Value

Figure 5-4. Key Fields of Various Record Types

All records within this hierarchical file have all key fields reproduced. In the case of Record Type A1, key 2 and key 3 have been inserted, but with the lowest value possible for the particular keys within the field. In the case of the End of Record Type B2, the highest values for key 2 and key 3 have been inserted, et cetera.

Reproducing all key fields ensures a complete file can be sorted later and the complete file will retain the same hierarchical file structure, but in a new order.

The End-of-Record type (ID code (2)) has the checksum totals in the "other fields" area. These checksum totals are software checks on the hardware's reading and writing of records. The checksum must be in the same format as the other fields contained in the Record Type (ID code 1). This checksum should contain such information as the number of records written for the particular record-type, a "hash" total for some of the other data items listed; e.g., employee numbers.

The size of all fields and the size of the record itself must be the same for all records within a given record type, but not necessarily the same size for records belonging to different record types. All records of type A1 must be the same size. All records of type B1 must be the same size. The size of non-key fields within the records of type A-1 should, in general, not be the same as the non-key fields within record type B1. Thus, the overall length of record type A1 and record type B-1 most likely is different.

Figure 5-5 is an illustration of a file containing the New England Power Company, with one company, Boston Edison; one division, Metro; one city, Cambridge; two streets, Inman and Mass Ave.; three lights on each street, i(1), i(2), i(3). The record types are labeled as follows.

A1	-	Company 02 - Division 03
B1	-	Division 03
C1	-	City/Town
D1	-	Street
E1	-	Light
E2	-	End of Lights on Street
D2	-	End of Streets in City
C2	-	End of Cities in Division
B2	-	End of Division in Company
A2	-	End of Companies in Utility System

Record Number	Record Type	Key 1	Key 2	Key 3	Key 4	Key 5	Other Information
1.	A1	Boston	0	0	0	0	
2.	B1	Boston	Metro	0	0	0	
3.	C1	Boston	Metro	Cambridge	0	0	
4.	D1	Boston	Metro	Cambridge	Inman	0	
5.	E1	Boston	Metro	Cambridge	Inman	1	
6.	E1	Boston	Metro	Cambridge	Inman	2	
7.	E1	Boston	Metro	Cambridge	Inman	3	
8.	E2	Boston	Metro	Cambridge	Inman	99999	
9.	D1	Boston	Metro	Cambridge	Mass Ave	0	
10.	E1	Boston	Metro	Cambridge	Mass Ave	1	
11.	E1	Boston	Metro	Cambridge	Mass Ave	2	
12.	E1	Boston	Metro	Cambridge	Mass Ave	3	
13.	E2	Boston	Metro	Cambridge	Mass Ave	9999	
14.	D2	Boston	Metro	Cambridge	ZZZZ	9999	
15.	C2	Boston	Metro	ZZZZ	ZZZZ	9999	
16.	B2	Boston	ZZZZ	ZZZZ	ZZZZ	9999	
17.	A2	ZZZZ	ZZZZ	ZZZZ	ZZZZ	9999	

Figure 5-5. New England Power Company File

5.6 LONG RECORDS

When a record is longer than 253 bytes, it is advisable to break the record into two or more smaller subrecords. One must be able to identify the parts of the complete record; both subrecords should have different record types, but the second subrecord's key field must be larger in value than the associated key field of the first subrecord. For example, Record 1 has a key field of Jones 1 and Record 2 has a key field of Jones 2. This allows sorting of both record types together -- as the single unit they logically are -- using their corresponding key fields.

Assume a file consists of three record types, S, R, T. Record type T is longer than 253 bytes. Therefore, record T must be broken into two shorter records which are renamed T and U. In order to maintain the subrecords in proper sequence, their associated keys are differentiated by the value of key 3b in record type U1 exceeding the value of key 3a in record type T1 (Figure 5-6). The two new records must be given different record ID codes, since their fields may be in different formats.

S1	Key 1	LV	LV	Other Fields
R1	Key 1	Key 2	LV	Other Fields
T1	Key 1	Key 2	Key 3a	Other Fields
U1	Key 1	Key 2	Key 3b	Other Fields

Key 3b must be greater than key 3a.

Figure 5-6. Key Fields in Subrecords

5.7 SAMPLE CODING

It is assumed the following file management utilities for tape input/output are available (refer to General Utilities manual).

		Initialize Volume
DEFFN'	240	Open Output File
DEFFN'	241	End-of-Volume/File Output
DEFFN'	250	Open File Input
DEFFN'	251	Close Volume Input

CAUTION: Maximum of one file per cassette tape.

Whenever an incorrect action is attempted, the utility causes Line 0 of the CRT to indicate an error has occurred. Line 1 prompts the operator as to what action to take. Line 2 is reserved for a response from the operator. The utility resumes once the correct course of action has been taken by the user. See Figure 5-7 and 5-8 for examples of the use of these five routines.

Initialize Cassette

A stand-alone utility program. All new cassette tapes to be used for storage of files first must be initialized with this utility, which writes special blocks needed for the proper execution of the remaining tape utilities. If the header blocks are not written, the execution of the remaining tape utilities result in a System 2200 execution error.

DEFFN' 240 Open Output File

Employed to open an output file before writing on a file for the first time. This utility requires the passing of the following parameters.

a. 8 character file name

b. file number associated with this file (a number from 1 to 3)

c. today's date (JULIAN = YYDDD)

d. retention period in days - the number of days the file must be kept before it may be destroyed; whenever this utlility is called, the header blocks are checked to ensure that a file is not destroyed inadvertently before the retention period has expired.

e. volume number (generally "one")

DEFFN' 241 End-of-Volume/File Output

Serves one of two purposes, depending upon which of two parameters is passed to it: either closes a volume of a multi-volume file, or closes a file.

End-of-Volume Output closes the current volume of the indicated file and requests the operator mount a new tape. The subroutine also automatically increments the volume number and opens the new volume after it is properly mounted. End-of-Volume Output must be employed upon reaching the maximum number of allowable data records on a given tape. For a 150-foot tape, the maximum number of data records is 300; for a 75-foot tape, the maximum number is 150 data records.

Before calling End-of-Volume Output, the programmer first must write a warning record in the same format as the next expected record. The warning record alerts the program an end-of-volume is approaching and the user must call the Close Volume Input utility (see DEFFN' 251).

End-of-File Output closes the current file on tape. Again, as in the End-of-Volume, a warning record must be written first.

A DEFFN' 240 (Open Output File) must be executed previously in order to open this file. The parameters originally used in the OPEN are saved. Hence, the only parameters that need be passed to DEFFN' 241 are:

a. File number (1 to 3)

b. "EOV" or "EOF", close volume, close file.

<u>DEFFN' 250 Open File Input</u>

Used to open a new file for reading previously written data. The parameters passed to this utility are file name, file number and volume number. If the procedure is not executed properly, the CRT notifies the operator of the error and issues the proper instructions.

<u>DEFFN' 251 Close Volume Input</u>

Always must be invoked after reading a warning record. It determines whether it is the end of a file or the end of a volume for a multi-volume file. If it is the end of a volume, the operator is requested to mount a new volume, and the new volume is opened by DEFFN' 251. A parameter is passed back to the program indicating whether it was the end of the volume or the end of a file. The file number must be passed to this program.

BASIC	Statement	Comments
140	Select #2, 10A	Associates a file number with a particular tape device address.
150	Q1 = 73206	Today's Julian date must be assigned to variable Q1. It more typically is assigned by invoking the accept date subroutine (see General Utilities).
160	GOSUB' 240 (file number, "file name", retention period in days, volume number)	Asks the user to mount a tape cassette, checks it is okay to use (valid file tape), and writes the proper header block. The parameter may be passed as immediate data, or as any variable arguments. They are listed below with the arguments as used in the subroutine. file number = numeric W file name = 8 character alphanumeric W$(W) retention period = numeric Q4(W) volume number = numeric Q5(W)
	. . . .	
220	A = 0	
230	DATASAVE #n, 'argument list'	Writes the actual records.
240	A = A + 1	Keeps a count of number of records written in A.
	. . . .	
250	If A < 300 THEN 230	If the number of records written (A) is less than 300, writes a new record; if not, then must end volume.
	. . . .	
260	DATASAVE #n, 'warning record'	Writes warning record.
270	GOSUB' 241 (file number, "EOV")	Writes an End-of-Volume record and requests the operator to mount a new volume then writes the proper header block on the new volume.
280	GOTO 220	Writes more data records.
	. . . .	
400	REM THIS SECTION PROCESSES END-OF-FILE	When finished recording the file, go to this section.
410	DATASAVE #n, 'warning record'	Writes warning record.
420	GOSUB' 241 (file number, "EOF")	Closes the file.

Figure 5-7. Writing Record

BASIC	Statement	Comments
1100	Select #2, 10A	Associates a file number with a particular tape device address.
1130	GOSUB' 250 (file number, "file name", volume number)	Tells the operator to mount the volume (cassette) and opens the file.
	. . . .	
1200	DATALOAD #n, 'argument list'	Reads the record of the file.
1210	IF 'argument list' = 'warning list' THEN 1400	
	. . . .	Process.
1250	GOTO 1200	
1400	GOSUB' 251 (file number)	Closes the current volume, and if end of volume, tells the operator to mount the next volume. Q5$ = "V" if end of volume; Q5$ = "F" if end of file
1410	IF Q5$ = "V" THEN 1200	Process more data or end of file.
1450	(logical end of file)	

Figure 5-8. Reading Record

CHAPTER VI

SCREEN USAGE TECHNIQUES

6.1 INTRODUCTION

The screen is the principal communication device between the program and the operator. A consistent personality must be presented on the screen at all times to avoid confusing the operator. Thus, the goals of screen usage techniques are:

- The operator always must see the same general layout on the screen, regardless of what program is being run.
- The dialogue used for program/operator interaction must be the same from program to program.

6.2 SCREEN LAYOUT

The screen consists of 16 rows numbered 0 through 15, and 64 columns numbered 0 through 63.

Input Phase

The normal procedure for the operator to start the first phase of any program (the input phase) is to key CLEAR CR/LF, LOAD CR/LF, RUN CR/LF. The program then must clear the screen entirely.

In the input phase, Line 0 of the screen is used for system messages, such as mounting tapes, changing paper, et cetera. Line 1 is used for dialogue to the operator in the form of "prompt" statements; such as, PLEASE INPUT TODAY'S DATE. Line 2 displays all data being entered by the operator. Line 3 indicates invalid information has been inputted. To recapitulate:

Line 0 is for system messages,

Line 1 is for prompts and prompts only,

Line 2 is for input required by prompt message, and

Line 3 is for input errors.

These four lines never are used for any other purpose in the input phase (see Table 6-1).

Table 6-1. First four Lines of CRT During Input Phase

LINE	DESCRIPTION	ACTUAL DISPLAY
0	System Messages	INVOICING
1	Prompt Messages	ENTER CUSTOMER NUMBER
2	Info requested by Line 1	98765
3	Input errors	INVALID CUSTOMER NUMBER

Execution Phase

During the execution phase, the screen must be erased and the phrase "EXECUTING THE PROGRAM" shown at the start of Line 0.

Output Phase

Three major ways of displaying output on a System 2200 are: on the screen; hardcopy only; and on the screen as a verification and then a hardcopy. The suggested procedure for each option follows:

Screen Only

1. Clear the screen using PRINT HEX (03).

2. Ask the operator whether or not the output is desired page by page, or line by line with a selected PAUSE interval. The option is selected by using the standard dialog format in lines 0 through 3. The operator must be able to view a new page of results by simply keying CR/LF.

3. When finished, clear the screen and display "END OF PROGRAM" in the center of the screen.

Hardcopy Only

1. Clear the screen.

2. Line 0 is reserved for system message to be sure the output devices are ready.

3. Line 1 is for further prompting; e.g., "IS THE PAPER POSITIONED CORRECTLY?"

4. Line 2 accepts the response.

5. When everything is verified, erase the screen; Line 0 must display "PRINTING".

Screen and Then Hardcopy

Follow the procedure in Screen Only, omitting Step 4. Instead of displaying "END", clear the screen and in Line 1 ask if a hardcopy is desired with the response on Line 2. If the answer is 'Yes', proceed as in Hardcopy Only. If the answer is 'No', use Lines 1 and 2 for further dialogue as needed.

6.3 SCREEN USAGE UTILITIES

Position Cursor

Sets position of cursor to any desired location (row, column) with an option to erase either the rest of the line or any number of the remaining lines of the screen.

Accept CRT Input

Setting the cursor and printing the message in the proper position for prompting on Line 1 and responding on Line 2.

Operator Wait

Displays a prompt message on Line 1 for the operator to key CR/LF.

These subroutines are explained in detail in the General Utilities manual.

(This page intentionally left blank)

CHAPTER VII

REPORT PRODUCTION CONVENTIONS

7.1 PRE-PRINTED CONTINUOUS FORMS

Provide for an alignment mark at the top left and top right of the form (see Figure 7-1) to aid the operator in properly inserting the form in the printer.

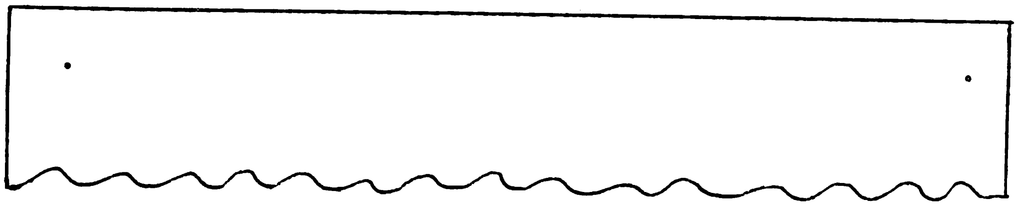


Figure 7-1. Alignment Marks

The next step in ensuring proper alignment of a form is to employ print masks. A print mask is a series of "X's" representing the alpha or numeric fields with the appropriate delimiters, commas, and periods. These X's, commas and periods represent the real data that is printed on a pre-printed form after program execution (see Figure 7-2).

Initially, one mask (consisting of a full page of data) should be printed. A dialog then is set up lines 0, 1, 2, 3 of the CRT to query the operator for repeating the display.

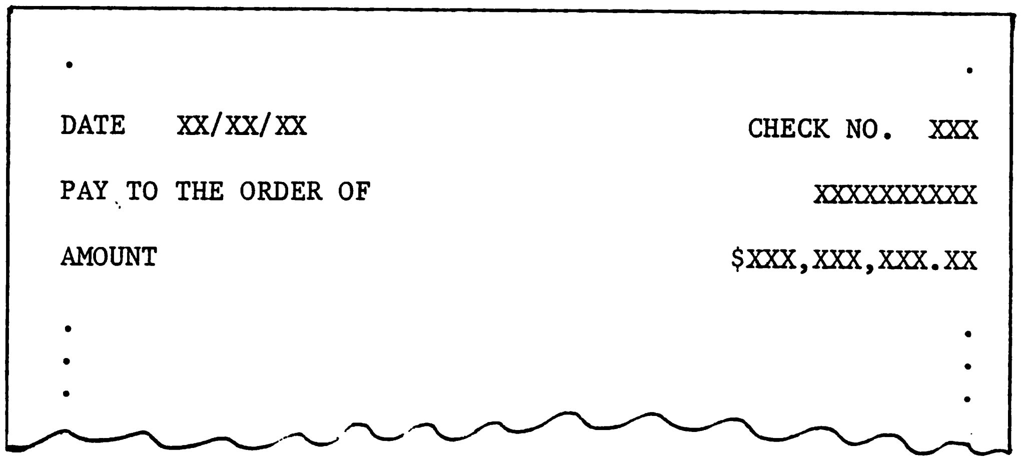


Figure 7-2. Sample Print Mask

7.2 STANDARD, NON PRE-PRINTED, FLAT PACK FORMS (66 LINES/PAGE, 132 CHARACTERS/LINE)

The normal page of the report (see Figure 7-3) has the characteristics:

1. Line one blank

2. Line two blank

3. Line three

a.	Column 10	Title of major key; i.e., "DIVISION"
b.	Center of page	Customer name
c.	Column 110	Date
d.	Column 122	Page

4. Line four

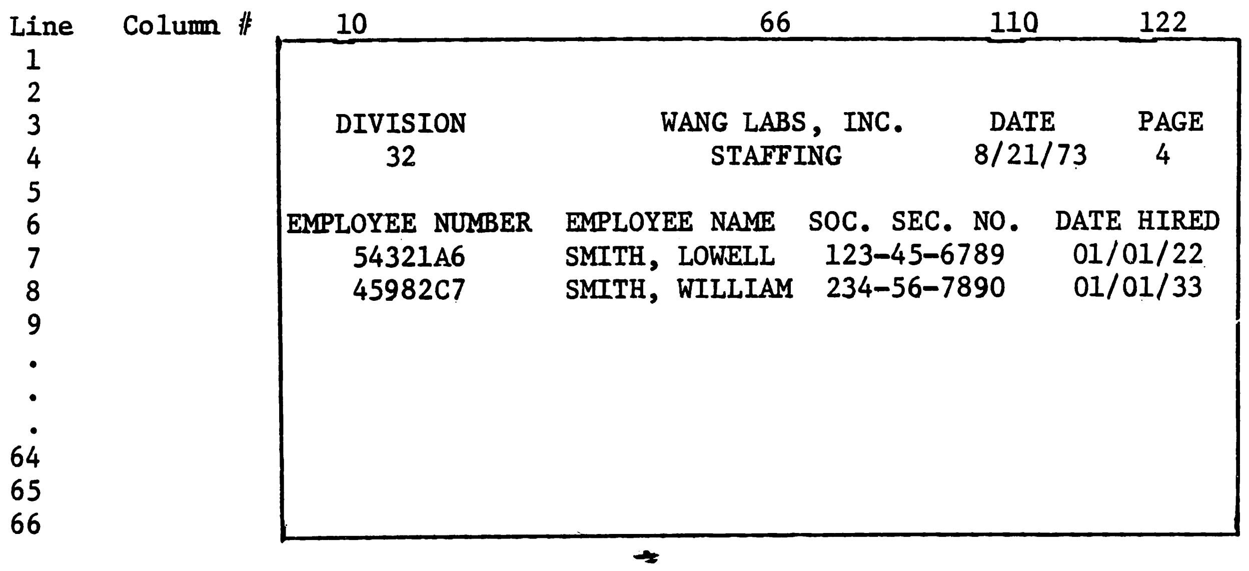
a.	Column 10	Major key DIVISION number
b.	Center of page	Report name
c.	Column 110	Date of report XX/XX/XX
d.	Column 122	Page number

5. Line 5 blank

6. Line 6 Column Headings

7. Line 7 (or as desired) first detail line. This line always must contain the complete identification of the item, even if the data is normally suppressed for subsequent entries with the same key values.

8. Successive detail lines up to line 66.



```
Line   Column #   10                      66               110      122
 1
 2
 3               DIVISION            WANG LABS, INC.     DATE     PAGE
 4                  32                  STAFFING        8/21/73     4
 5
 6              EMPLOYEE NUMBER   EMPLOYEE NAME   SOC. SEC. NO.   DATE HIRED
 7                  54321A6       SMITH, LOWELL     123-45-6789    01/01/22
 8                  45982C7       SMITH, WILLIAM    234-56-7890    01/01/33
 9
 .
 .
 .
64
65
66
```

Figure 7-3. Normal Report Page Form

The final line printed on the report, after as many pages as needed are used, contains the words "END OF REPORT", left justified.

(This page intentionally left blank)

CHAPTER VIII

DOCUMENTATION STANDARDS

8.1 INTRODUCTION

The two levels of documentation are system and program. The system documentation is a package of information designed by an analyst containing all the relevant information for a programmer to code the application; it is contained in the Systems Manual. The programmer produces further, more basic documentation to actually implement the programs within the system, contained in a Program Run Book. The Program Run Book includes such things as code listings, intermediate level flowcharts, et cetera. The complete system documentation, therefore, consists of one Systems Manual and an individual Program Run Book for each computer program.

8.2 SYSTEM LEVEL DOCUMENTATION

The documentation of system specifications is contained in a Systems Manual, including the following items.

Management Summary

Outlines the scope and the objective of the project. Explains the Systems Flowchart by briefly describing the inputs and outputs, and program functions. In addition, it lists the principal features and benefits derived from the features.

System Flowchart

Diagrams the flow of the system with sufficient detail to identify all inputs, outputs, and computer programs within the system. Input documents and output reports are named and numbered.

Machine Requirements

Specifies the equipment configuration for which the system is designed.

Operating Instructions

For each program, supply the following information:

a. Procedure for initial load of program, including special run-time processing option; e.g., how to run an end-of-month pass instead of a normal daily pass of the data.

b. Transaction list - a list of each input transaction processed by the program, including its method of initiation, normal dialog and description; i.e., functions it performs.

c. Messages produced by program other than those used to generate input, including error messages, cause of message, and resultant action to be taken.

d. Report control procedures - balancing and verification of accuracy of reports, normal handling and distribution of report.

e. Procedure for restarting program

Changes to the operating instructions must be duplicated in the appropriate Program Run Book.

Estimation of Throughput

A table must be produced estimating the overall throughput of the system, using average and peak volumes of the files, input, and reports detailed below.

Input Layouts

All dialog and screen usage for input data must be documented on the CRT layout form. Samples of forms that contain the information to be inputted via the CRT/keyboard must be provided.

Report Layouts

All reports, both CRT and hardcopy, must be documented on the appropriate forms.

File Layout

All files used in the system to store data, even temporarily, must be documented on the file layout form.

8.3 PROGRAM LEVEL DOCUMENTATION

For each program within a system, a Program Run Book is maintained. The following items are to be included in the Program Run Book.

Narrative Description of Program

A brief overview of the purpose and functions of the program, including special features. The flow is described in the same sequence as the Logic Flowcharts, although in a much more abbreviated form. The overview is equivalent to the management summary in the System Level documentation.

Special Explanatory Information

Includes any of the following types of items, if used: decision tables; data tables; formulas; codes; et cetera.

Logic Flowcharts

Intermediate level flowcharts indicating the general operational processes and logic flow of the program. An expansion of the program description given in the overview. While not a one-for-one relationship with the detail coding, it should contain sufficient tags or line numbers to point to the section in the source code containing the routine described.

Operating Instructions

Identical to the operating instructions in System Level documentation for each program. Changes must be duplicated in the Systems Manual.

Document and File Layouts

Describe each input and output file and output report design. May be the same information contained in the Systems Manual. It is included so each Program Run Book contains a complete description of the program.

Source Program Listing

Hardcopy printout containing the source coding used by the program. Most recent version only.

8.4 DOCUMENTATION FORMS

(This page intentionally left blank)

SYSTEM ______________ PROGRAM ______________

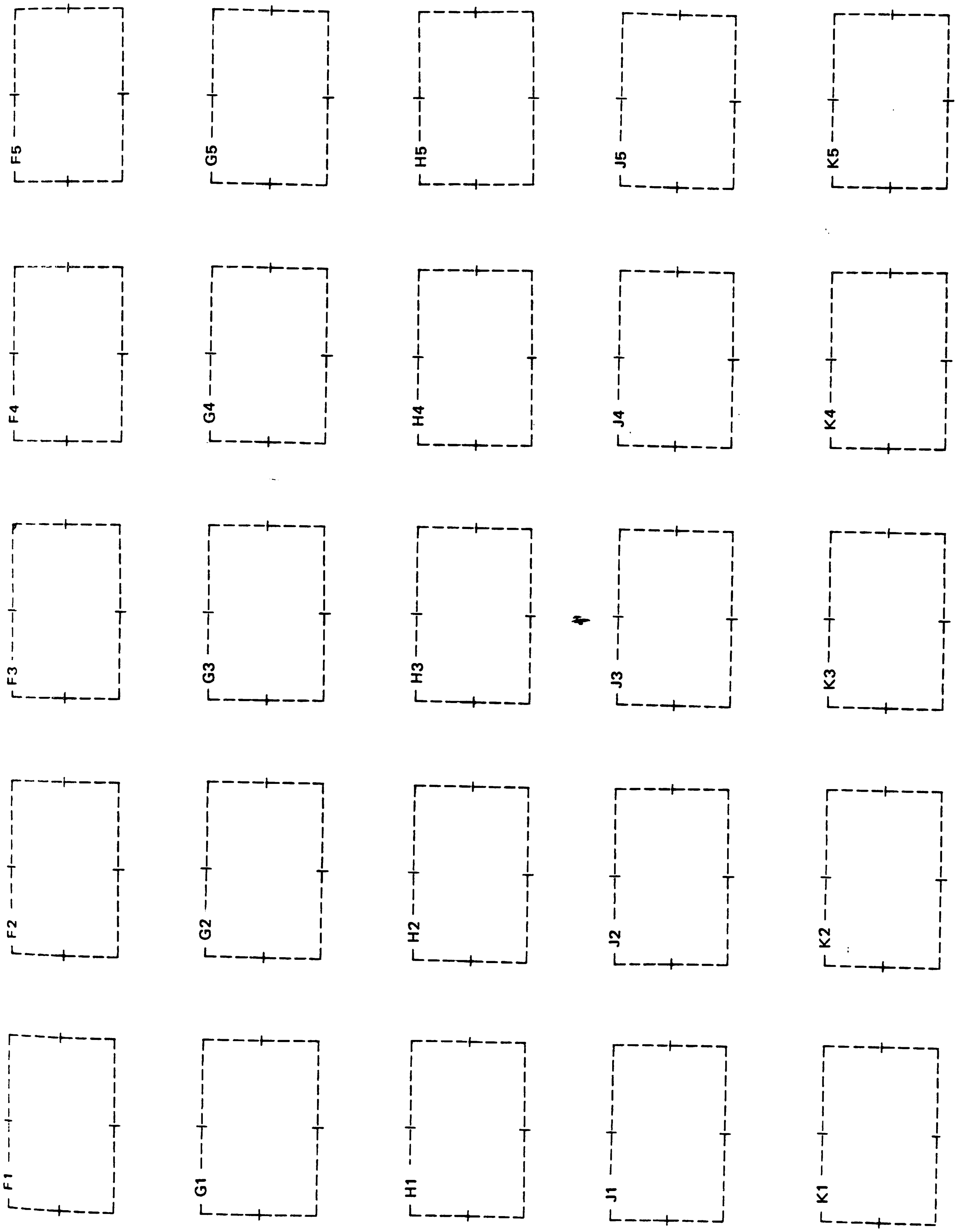
VERSION ______________ PREPARED BY ______________ DATE ______________

CHART ______________

SHEET ______ OF ______

A1	A2	A3	A4	A5
B1	B2	B3	B4	B5
C1	C2	C3	C4	C5
D1	D2	D3	D4	D5
E1	E2	E3	E4	E5

← Fold under at dotted line.



Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

← Fold under at dotted line.

SYSTEM NAME ______________________ DATE ______________

PROGRAM NAME ______________________ VERSION ______________

TAG	STMT NO.	BASIC STATE
	10	
	20	
	30	
	40	
	50	
	60	
	70	
	80	
	90	
	00	
	10	
	20	
	30	
	40	
	50	
	60	
	70	
	80	
	90	
	00	
	10	
	20	
	30	
	40	
	50	
	60	
	70	
	80	
	90	
	00	

SYSTEM 2200 CODING FORM

PROGRAMMER PAGE____ OF____

ENT

DESTINATION TAG

WANG LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL (617) 851-4111, TWX 710 343-6769, TELEX 94 74.1

Printed in U.S.A.
700-3180A
1-74-1M
Price $2.50

SYSTEM 2200 RECORD DESCRIPTION

Page ___ of ___

SYSTEM NAME ____________________

PROGRAM NAME ____________________ VERSION ____________________

FILE NAME ____________________ DATE ____________________

RECORD ____________________

PROGRAMMER ____________________

FIELD NAME	TYPE	VAR.	ELEM LEN.	TAPE LEN.	KEY	NOTES
RECORD ID CODE						

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL (617) 851-4111, TWX 710 343 6769, TELEX 94 7421

Printed in U.S.A.
700-3182
9-73-2.5C
Price 1.50

SYSTEM 2200 VARIABLE CHECK-OFF LIST

PROGRAM NAME ______________________ DATE ______________

VERSION __________ __________ PROGRAMMER ______________

SYSTEM ____

NUMERIC SCALARS
FORMAT = MN

N \ M	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Δ																										
1																										
2																										
3																										
4																										
5																										
6																										
7																										
8																										
9																										
0																										

NUMERIC ARRAYS
FORMAT = MN(

N \ M	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Δ																										
1																										
2																										
3																										
4																										
5																										
6																										
7																										
8																										
9																										
0																										

ALPHA NUMERIC SCALARS
FORMAT = MN$

N \ M	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Δ																										
1																										
2																										
3																										
4																										
5																										
6																										
7																										
8																										
9																										
0																										

ALPHA NUMERIC ARRAYS
FORMAT = MN$(

N \ M	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Δ																										
1																										
2																										
3																										
4																										
5																										
6																										
7																										
8																										
9																										
0																										

NOTE:
0 = NON COMMON
1 = COMMON DEFINED BY THIS MODULE
2 = COMMON DEFINED BY PREVIOUS MODULE

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

Printed in U.S.A.
700-3175
9-73-2.5C
Price $1.50

SYSTEM 2200 NUMERIC ARRAYS

PROGRAM NAME ______________________ VERSION ______________

SYSTEM ______________________ DATE ______________

PROGRAMMER ______________________________________

VARIABLE	DESCRIPTION

WANG LABORATORIES, INC.
836 NORTH STREET TEWKSBURY MASSACHUSETTS 01876 TEL (617) 851-4111 TWX 710 343 6769 TELEX 94 7421

Printed in U.S.A.
700-3177
9-73-2.5C
Price $1.50

SYSTEM 2200 NUMERIC SCALARS

PROGRAM NAME ______________________ VERSION ______________

SYSTEM ______________________ DATE ______________

PROGRAMMER ______________________

VARIABLE	DESCRIPTION

Printed in U.S.A.
700-3178
9-73-2.5C
Price $1.50

SYSTEM 2200 ALPHA-NUMERIC SCALARS

PROGRAM NAME ______________________ VERSION ____________

SYSTEM ______________________ DATE ____________

PROGRAMMER ______________________________________

VARIABLE	DESCRIPTION

SYSTEM 2200 ALPHA-NUMERIC ARRAYS

PROGRAM NAME ______________________ VERSION ______________

SYSTEM ______________________ DATE ______________

PROGRAMMER ______________________________________

VARIABLE	DESCRIPTION

(This page intentionally left blank)

PROGRAM NAME ____________ VERSION ____________ PAGE ____

DATE ____

SYSTEM NAME ____________

PROGRAMMER ____________

CHART TITLE ____________

Fold here

SYSTEM 2200 PRINTER SPACING CHART

64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132

Fold here

WANG LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876 TEL (617) 851-4111, TWX 710 343 6739, TELEX 94 7421

Printed in U.S.A.
700-3181A
2-74-5C
Price $2.50

SYSTEM 2200

SYSTEM ____________________

PROGRAM NAME ____________________

MODULE ____________________ VERSION ____________________

	0					5					10					15					20					25					30
0																															
1																															
2																															
3																															
4																															
5																															
6																															
7																															
8																															
9																															
10																															
11																															
12																															
13																															
14																															
15																															

0 5 10 15 20 25 30

Note: During input phase, line 0 = SYSTEM MESSAGE; 1 = OPERATOR PROMPT;
2 = DATA INPUT; 3 = ERROR MESSAGE. Lines 4-15 = display user data.

;CREEN LAYOUT

________ REPORT/OPERATION ________ PAGE ____ of ____

________ DATE ________

____ PREPARED BY ________

35 40 45 50 55 60 63

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

35 40 45 50 55 60 63

Printed in U.S.A.
700-3187A
2-74-5C

INDEX

	PAGE
Accept CRT Input	6-3
Application Header Record	5-5
Cassette, Initialization Of	5-15
Classes Of Utilities/Subroutines	1-1, 1-2
Cursor, Position Of	6-3
Data Record Design Techniques	5-8 to 5-10
DEFFN' 240 Open Output File	5-15
DEFFN' 241 End-Of-Volume/File Output	5-15, 5-16
DEFFN' 250 Open File Input	5-16
DEFFN' 251 Close Volume Input	5-16
Document And File Layouts	8-3
Documentation Forms	8-3 to 8-17
Documentation, Program Level	8-2, 8-3
Documentation, System Level	8-1, 8-2
End-Of-Volume/End-Of-File Record	5-5 to 5-7
Estimation of Throughput	8-2
Execution Phase	6-2
Fifth And Sixth Characters (Program Name)	3-1
File	2-1, 2-2
File Layout	8-2
File Name	3-2
File, Reading Of	5-7
File, Writing Of	5-7
First Three Characters (Program Name)	3-1
Flat Pack Forms (Non Pre-Printed)	7-2, 7-3
Format	4-4, 4-5
Fourth Character (Program Name)	3-1
Header And Trailer Records	5-5 to 5-7
Header/Trailer Conventions	5-7
Identification Codes	5-3, 5-4
Identification, Program Structure	4-1 to 4-3
Identification, Segmented Module	4-3
Input Layouts	8-2
Input Phase	6-1, 6-2
Introduction, Definitions	2-1
Introduction, Documentation Standards	8-1
Introduction, File Structure	5-1, 5-2, 5-3
Introduction, Names	3-1
Introduction, Program Structure	4-1
Introduction, Programming Tools	1-1
Introduction, Screen Usage Techniques	6-1

	PAGE
Key Fields	5-11 to 5-13
Logic Flowcharts	8-3
Logical Records, Size Of	5-8
Long Records	5-14
Machine Requirements	8-1
Management Summary	8-1
Module	2-1
Operating Instructions (Program)	8-1, 8-2, 8-3
Operator Wait	6-3
Output Phase	6-2, 6-3
Pre-Printed Continuous Forms	7-1
Program Level Documentation	8-2, 8-3
Program Name	3-1, 3-2
Program, Narrative Description Of	8-2
Program, Sample	4-5, 4-7
Programs	2-1
Record Description Form	5-10
Record Type Indicator	5-8
Record Types	5-6
Record, Reading Of	5-18
Record, Writing Of	5-17
Records, Header And Trailer	5-5 to 5-7
REM Statement	4-4, 4-5
Report Layouts	8-2
Reports	2-2
Resident Utilities	1-1
Sample Coding	5-14 to 5-18
Screen Layout	6-1 to 6-3
Screen Usage Utilities	6-3
Seventh And Eighth Characters (Program Name)	3-2
Source Program Compression	4-8
Source Program Listing	8-3
Stand-Alone Utilities	1-1
Subroutines	1-2
System	2-1
System Flowchart	8-1
System Level Documentation	8-1, 8-2
Utilities/Subroutines, Classes Of	1-1, 1-2
Variables	3-3 to 3-8
Warning Record	5-5

WANG LABORATORIES (CANADA) LTD.
180 Duncan Mill Road
Don Mills, Ontario M3B 1Z6
TELEPHONE (416) 449-7890
TELEX 06-966546

WANG EUROPE, S.A.
Buurtweg 13
9412 Ottergem
Belgium
TELEPHONE: 053/74514
TELEX: 26077

WANG ELECTRONICS LTD.
1, Olympic Way, 4th Floor
Wembley Park,
Middlesex, England
TELEPHONE: 01/903/6755
TELEX: 923498

WANG FRANCE SARL
47, Rue de la Chapelle
Paris 18, France
TELEPHONE 203.27.94 or 203.25.94

WANG LABORATORIES GMBH
Moselstrasse No. 4
6000 Frankfurt am Main
West Germany
TELEPHONE (0611) 252061-64

WANG SKANDINAVISKA AB
Fredsgaten 17
S-172-23
Sundbyberg 1, Sweden
TELEPHONE 08-98-12-45

WANG NEDERLAND B.V.
Damstraat 2
Utrecht, Netherlands
TELEPHONE 030-930947

WANG PACIFIC LTD.
61, King Yip Street, 1st Floor
Kwun Tong Kowloon, Hong Kong
TELEPHONE 3-434231/2

WANG INDUSTRIAL CO., LTD.
110-118 Kuang-Fu N. Rd.
Taipei, Taiwan
Republic of China
TELEPHONE 784181-3

WANG GESELLSCHAFT MBH
Grinzinger Allee 16
1190 Vienna 19
Austria
TELEPHONE (0222) 32.42.43

WANG COMPUTER PTY. LTD.
25 Bridge Street
Pymble, NSW 2073
Australia
TELEPHONE 449-6388

WANG INTERNATIONAL TRADE, INC.
836 North Street
Tewksbury, Massachusetts 01876
TELEPHONE (617) 851-4111
TWX 710-343-6769
TELEX 94-7421

WANG COMPUTER SERVICES
836 North Street
Tewksbury, Massachusetts 01876
TELEPHONE (617) 851-4111
TWX 710-343-6769
TELEX 94-7421

24 Mill Street
Arlington, Massachusetts 02174
TELEPHONE (617) 648-8550

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3199B
6-74-1.5M