

2200 DEVELOPMENT STANDARDS AND GUIDELINES

Release 1.0

**Prepared by:
Quality Assurance
January 1979**



LABORATORIES, INC.

ONE INDUSTRIAL AVENUE, LOWELL, MASSACHUSETTS 01851, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

NOTICE:

The staff of Wang Laboratories, Inc., has taken due care in preparing this document. Nothing contained herein modifies or alters in any way the standard terms and conditions of the Wang purchase, lease, or license agreements, nor increases in any way Wang's liability to the customer. In no event shall Wang Laboratories, Inc., or its subsidiaries be liable for incidental or consequential damage in connection with or arising from this document or related documents.

© Wang Laboratories, Inc., 1979

TABLE OF CONTENTS

	Page
SECTION 1 Introduction	1
SECTION 2 2200 Standards and Guidelines	2
2.1 Program File Names	3
2.2 Variables	3
2.3 Source Program	4
2.4 Compressed Program/Disk	6
2.5 Subroutines	6
2.6 Special Function Keys (SF)	7
2.7 Data File Names	7
2.8 Record Description Forms	7
2.9 Key Fields	7
2.10 Data Save End	7
2.11 Screen Displays	8
2.12 Menus	8
2.13 Printer Output	9
2.14 Printer Usage	9
2.15 Disk Usage	9
2.16 Special MVP Considerations	10
2.17 Documentation	11
2.18 Testing	12
SECTION 3 Implementation of Standards and Guidelines	13
3.1 Narrative	
3.2 Development Flow	
SECTION 4 Standard/Guideline Request	14
4.1 Overview	15
4.2 Request Form	16
APPENDIX A Examples	17
APPENDIX B Standards Check-off List	29
APPENDIX C Distribution	31

SECTION 1 INTRODUCTION

The following document is an attempt to capture the software development style for the Wang 2200 series. It is also intended to be a point of reference for establishing related documents in the future.

This document, although comprehensive in nature, covering the subject of 2200 standards and guidelines, is designed with enough flexibility to encourage creative programming. Extreme care has been taken in both the wording and categorizing to promote effective practical standards.

An important distinction has been made between a "Standard" and a "Guideline":

A Standard by our definition is a definite rule or principle established by general consent and authority.

A Guideline, on the other hand, is a recommended policy or procedure.

Both are intended to ensure consistency and quality in the software product that we produce (refer to Section 4 for more detail).

In summary, this document hopes to introduce the style and organization of our work in a relatively short period of time. The usefulness and success of this, however, will depend greatly on a genuine cooperative effort by all those involved in 2200 software development.

Quality Assurance

SECTION 2
STANDARDS AND GUIDELINES

SECTION 2 2200 STANDARDS AND GUIDELINES

2.1 Program File Names

Standards:

- A. The starting module must be cataloged under the program file name of "START." This convention will alleviate the question of what module to load when starting a system. This also utilizes the "LOAD RUN" feature in both the 2200 VP and 2200 MVP.
- B. The file name of a menu program consists of eight positions with the first four positions indicating the system or menu function, and the last four characters containing "MENU" (e.g., "A/R MENU" and "PAYRMENU" for accounts receivable and payroll menus respectively). If several menus are used in a system, "MEN1," "MEN2," etc., could be used.
- C. All other program file names consist of exactly eight positions. The first four characters correspond to the system name. The fifth, sixth, and seventh positions represent the program number within the system, starting with 10 and increments of 10 whenever possible. The last position contains a letter that designates the module within a program (e.g., the second module of a program in a payroll system could contain a name "PAYR020B") (see Appendix A, Figure 10).

2.2 Variables

Standards:

- A. A description of what each variable represents must be documented in the source listing of each module with the exception of standard, systemwide utility variables (see Appendix A, figure 2).
- B. A variable check-off list must be filled out and included with the documentation (see Appendix A, Figure 1).
- C. Variables Q through W are reserved for ISS utilities.
- D. Switches must be consistent. The ON and OFF switches must contain the value of 1 and 0, respectfully (i.e., 1 = ON, 0 = OFF).

Consistent variable usage must be maintained throughout the system.

- E. Documentation must clarify the distinction between common and non-common variables, and specify which variables remain common after each Program overlay.

Guidelines:

- A. If speed of execution is a significant consideration then the most frequently used variables should be dimensioned or commoned last to maximize processing efficiency on the 2200T processor (this is immaterial on a 2200VP/MVP).

- B. To distinguish between different usages of variables, a grouping scheme should be employed and documented in the main-line source code (e.g., the first character position could distinguish flags/switches, counters, working variables and record variables, while a range of numeric characters in the second position could distinguish common and non-common variables)(see Appendix A, Figure 2).
- C. Alphanumeric String variables and numeric variables should appear similar for related items (e.g., C9\$ = "2200 calculator costs = \$ " ; C9 = 3500.00).
- D. Scratch or work variables should be used whenever possible to reduce memory requirements.

2.3 Source Program

Standards:

- A. Program statement numbers are in increments of 10. However, with an overlaying scheme this may not be possible.
- B. Every module has a line 10 REM. This statement should contain the module file name, version number, (module date), copyright and (optional description) (see Appendix A, Figure 3). This format must be in all main-line modules and all stand-alone programs. Sublevel modules need not include copyright but must contain the name, version, and module date.

1. Version Numbers - All new projects will begin with Version 1.0, where left of the decimal point reflects the "RELEASE" number and right of the decimal point reflects the "REVISION" number. Collectively the number reflects the current version of that system or module.

"Revision Numbers" - will be assigned to the project when maintenance of a system is deemed necessary, but the changes are relatively minor. The revision number must be incremented by 1 in the module being revised and in the "START" module (e.g., 1.1).

"Release Numbers" - will be incremented by one and revision number reset to zero (e.g., 2.0) when a major change is to be made to the system (e.g., design change, major enhancement).

2. Module Date - The initial date for all new projects reflects a programming completion date for that system. If a "REVISION" is necessary, the line 10 REM of the module affected, Start module, and Primary menu will be updated with the date of revision. The date should be enclosed within parentheses.
3. Copyright - The following must appear in the start module, the highest level menu, and all stand-alone programs:

THIS PROGRAM IS A COPYRIGHT PRODUCT OF
WANG LABORATORIES, INC., UNAUTHORIZED
REPRODUCTION OR USE IS PROHIBITED.

4. Optional Description - A short description describing the module function.
- C. Source and Compressed module names must be the same, with the exception of pooling subroutines together under a common file name. This must be well documented in the source.
 - D. Source statements must not branch to a REM, or follow a REM on the same line (e.g., 200 A = K: REM RESET POINTER: GO TO 100).
 - E. Source code contains one statement per line, except when REM's are used as a second statement for comments (e.g., 200 ON ERROR E\$, B\$ GO TO 900:REM CHECK DATA LOAD DC OPEN FOR STATION FILE).
 - F. All program maintenance is performed with source code.
 - G. The "START" module is a loader module only. This minimizes the effort involved in recoding a system "START" module to be incorporated into a multisystem menu. (e.g., a user might want to place ISS Utilities, Text Editing, and T/C Utilities on a disk which would be accessed from one start module.)
 - H. Document source code with REM's whenever possible.

Guidelines:

- A. Defaults should be used whenever possible. In some cases, however, a required response is warranted, and no defaults should be allowed.
- B. Certain Statement lines should be indented for legibility, especially FOR/NEXT loops.
- C. Modular programming structure should be used.
- D. A programming convention of "REM---" for subroutine boundary/ identifiers and "REM****" for main line code, aids in identifying portions of the system.
- E. To avoid the loading restriction problem of what device number a user is loading from, all application programs should load modules using T#0, (e.g., 300 LOAD DC T #0, "PAYRO10A," Note ISS uses T#0). The user can key in the immediate mode a SELECT DISK 320 (or whatever address he/she desires); thus, systems can be run from any address.

2.4 Compressed Program/Disk

Standards:

- A. All source code must be compressed for release.
 1. T Processors - Compression must be restricted to 180 characters per line on all system software. This is due to editing restriction on the T Processor.
 2. VP/MVP Processors - There is no editing restriction on these processors. Compression of 256 characters per line must be performed for maximum storage utilization.
- B. Each module should have extra sectors for future maintenance if possible. This is left up to the discretion of the project leader to specify the amount of extra sectors each file contains. The traditional guideline is 2 extra sectors.
- C. Compressed disks should contain only files pertinent to the particular system (i.e., no scratch or irrelevant files). However, a particular system may require "TEST" files for demonstration purposes.

2.5 Subroutines

Standards:

- A. DEFFN' 200 through '255 are reserved for ISS Utilities.
- B. If an ISS subroutine is modified, its DEFFN' 200 - '255 number is changed to DEFFN' 100 - '155 respectively.
- C. DEFFN '0 through '31 are reserved for special function keyboard entry.

Guidelines:

- A. Standard subroutines should be used in their complete form whenever possible. Keep in mind they are designed and written for general use and may require modification for a specific application. If there is difficulty integrating them into your routines, see the supporting programmer concerning changes.
- B. To maximize efficiency on a 2200T processor, frequently used subroutines should be loaded at the top of memory (i.e., low line numbers) while infrequently used subroutines are loaded at the bottom of main line code. Subroutines that will remain in memory from one overlay to the next are handled more easily at the bottom of main line code.

2.6 Special Function Keys (SF)

Standards:

- A. SF '15 or '31 returns the user to a menu one level higher. The use of a HALT/STEP or RESET key prior to SF '15 or '31 may be necessary depending on the system.
- B. When the user is prompted for a selection of responses, all keys including Special Function Keys other than SF '15 or '31 that do not relate to the user options must be locked out.

2.7 Data File Names

Standards:

- A. A data file must be created so that its name appears in the catalog index.
- B. Data file and key file names consist of exactly eight positions. The first four of these should correspond to the name of the system (e.g., PAYR for payroll). The fifth position should read F for KFAM files or H for HFAM files, and the last three positions are used for a numbering sequence.

Guideline:

- A. Files other than KFAM or HFAM may utilize the fifth position to further describe the name. The last three characters could be used as a numbering scheme as in the program file name convention (e.g., "PAYRS010" is a sequential data file for a payroll system).

2.8 Record Description Forms

Standard:

- A. A record description form contains a record layout for data files and must be included with the documentation (see Appendix A, Figures 4 and 5).

2.9 Key Fields

Guideline:

- A. The primary key field in KFAM and HFAM files should be the first field within a record.

2.10 Data Save End

Guideline:

- A. The Data Save DC End or its equivalent (i.e., MUX END) should be used when applicable to assure that Copy/Verify and Reconstruct Index can handle data files properly.

2.11 Screen Displays

Standards:

- A. When switching from one function to another in execution, the CRT screen is completely cleared. An exception to this rule is when information in the screen is valuable for the next function (e.g., after entering default values for the ISS Utilities default option screen, the utility will display a "MOUNT INPUT DISK" prompt while retaining the current screen display for referencing the proper disk address).
- B. Dialogue throughout the system is done consistently.
- C. The screen should not be left blank for any significant period of time. If there is an excessive delay, a display must appear describing the activity (see Appendix A, Figure 7).

Guidelines:

- A. Responses should be enclosed in parentheses along with the default option (e.g., DATE O.K? (Y or N, DEFAULT = Y)).
- B. The programmer should avoid extraneous prompts whenever possible (e.g., if the operator responds 'N' to the prompt HARD COPY? (Y/N), the program can automatically assume the CRT as output without the display of the prompt CRT OUTPUT? (Y/N)).
- C. The CRT layout for operator response reads as follows:

LINE 0	System Messages
LINE 1	Prompts
LINE 2	Input required by prompt message
LINE 3	Input errors

(See Appendix A, Screen Usage Techniques.)

- D. The programmer should use screen layout forms to aid in designing a screen layout (see Appendix A, Figure 9).
- E. MVP Programs should minimize the amount of screen interaction to increase system efficiency.

2.12 Menus

Standards:

- A. The title of a system and the version numbers are displayed in the first line of the main menu (see Appendix A, Figure 8).
- B. The selection of functions in a menu are performed with Special Function Keys.

2.13 Printer Output

Standards:

- A. *Each multipage report must contain a page-numbering system.*
- B. *A report should be formatted in such a way as to be consistent and reasonable with the application.*

Guidelines:

- A. *Each report should contain the date in the form MM/DD/YY.*
- B. *A heading should use expanded print (PRINT HEX(OE); A\$).*

2.14 Printer Usage

Standards:

- A. *A check for printer selection is always made prior to printing.*
- B. *Special forms must have a print mask for lining up the form in the printer.*
- C. *When using standard stock, a HEX(ODOC) should be executed at the beginning of a report to clear the print buffer and skip to top of form.*
- D. *\$OPEN and \$CLOSE will be used for MVP applications to hog or release any printer (terminal or System).*
- E. *Printer device selection should be performed in a subroutine, so that software can be easily adapted from one hardware configuration to another.*

2.15 Disk Usage

Standard:

- A. *Disk hog mode will be used only when essential.*

Guidelines:

- A. *Efforts should be made to reduce disk head movement when planning disk access.*
- B. *Disk device selection should be performed in a subroutine, so that software can easily adapt from one hardware configuration to another.*

2.16 Special MVP Considerations

Standards:

- A. *Frequently used subroutines will be made part of a global partition.*
- B. *Foreground/background processing will not require immediate mode commands (i.e., \$RELEASE TERMINAL).*
- C. *Timing loops will not be used (i.e., timing of prompts or CRT displays).*
- D. *Menus should verify the existence of required global partitions.*
- E. *Data entry will be accomplished with INPUT/LINPUT. The use of KEYIN will be restricted to menus, (where SF Keys are used for program selection), or in cases where absolutely necessary.*

Guidelines:

- A. *Error traps should be appended to lines where errors are likely to occur rather than using ON ERROR Logic. This will promote better error recovery.*
- B. *Terminals can be remote and should be considered to reduce the amount of communication/cooperation required between them.*

2.17 Documentation

Standards:

- A. *Minimum hardware requirements.*
- B. *Range of hardware possibilities.*
- C. *Any other software that interfaces with the package.*
- D. *Describe each functional operation; for each include input, output, and a description.*
- E. *Description and layout for all Data files (see Appendix A, Figure 5).*
- F. *Explanation of how to load and operate the system including all options.*
- G. *Explanation of all error messages including conditions and recovery procedures.*
- H. *Narratives for all programs.*
- I. *System Flow Chart showing logical interaction of programs.*
- J. *Report layouts including CRT, PRINTER, and MENUS.*
- K. *Basic Screen Layouts (see Appendix A, Figure 8).*
- K. *Definition of terms and formulas.*
- L. *Variable check-off list (see Appendix A, Figure 1).*

2.18 Testing

Standards:

- A. *Run all options of all programs. If this is not possible due to their number, run all the most common options and choose at random from the rest.*
- B. *Observe the flow of logic through programs as they are normally used on sequence (e.g., ENTER DATA, SORT FILE, and PRINT REPORT).*
- C. *Check structure of all files accessed on the chain. Make sure they conform to documentation.*
- D. *Run any summary or month-end processing.*
- E. *Review output contained in all reports. Check that it is accurate, presentable, and in conformance to documentation. Similarly for CRT displays.*
- F. *Assure that all programs run in minimum core. Test the full range of configurations allowable for the system when necessary.*
- G. *Document all testing procedures, describing which options and sequences of options were tested (test spec). Save any test files and test output for future use.*

SECTION 3
IMPLEMENTATION OF STANDARDS AND GUIDELINES

TO BE DEVELOPED ----

SECTION 4
STANDARD/GUIDELINE REQUESTS

SECTION 4 STANDARD/GUIDELINE REQUESTS

4.1 Overview

2200 standards originated from informal guidelines developed by the earlier 2200 programmers. These guidelines were practical, effective, and were soon accepted as standard Programming Procedures. As new and more sophisticated programming techniques developed, many new suggestions were introduced. Some of them were included as part of the 2200 standards while others were not. Many, however, have remained as guidelines describing and recommending a programming development style. Together, these standards and guidelines have established a level of consistency and reliability in the software products being produced.

Today the Quality Assurance Group monitors and disseminates Standard/Guideline information throughout the development groups. But it is the responsibility of every group to strive for practical and effective approaches to software development that will allow for creativity as well as consistency and quality. It is with this in mind that the Standard/Guideline Request Form was developed.

The following page illustrates the form to be used in notifying the Quality Assurance Group of any ideas and suggestions that may contribute to Standards and Guidelines. These forms will be reviewed on a regular basis and responded to accordingly.

Quality Assurance

STANDARD/GUIDELINE REQUEST FORM

NAME: _____ EXTENSION: _____ DATE: _____

DEPT. #: _____ GROUP: _____ LOCATION: _____

Please note your suggestion or idea below. Use the opposite side of this form if additional space is needed.

I feel my suggestion would best benefit as a:

- STANDARD GUIDELINE THOUGHT FOR REVIEW
 OTHER

Send to:

Quality Assurance
Software Development
Dept. #93
Lowell

APPENDIX A EXAMPLES

Wang BASIC Variable Check-off List

PROGRAM NAME _____ DATE _____

VERSION _____ PROGRAMMER _____

SYSTEM _____

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 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\$(

NOTE:
0 = NON COMMON
1 = COMMON DEFINED BY THIS
MODULE
2 = COMMON DEFINED BY PRE-
VIOUS MODULE

(FIGURE 1)

IDENTIFICATION SECTION OF A PROGRAM

```
10 REM PAYR010A, 02.00 (1/15/78), (DEDUCTION CALCULATION)
20 REM MODULE NAME = PAYR010A
30 REM RELEASE = 02
40 REM REVISION = 00
50 REM DATE LAST CHANGE = 1/15/78
60 REM SYSTEM NAME = PAYROLL ACCOUNTING
70 REM PROGRAM NAME = DEDUCTION CALCULATION
80 REM MODULE NAME = COMPUTER FEDERAL TAX
90 REM THIS PROGRAM IS PART OF A GENERALIZED
100 REM APPLICATION SYSTEM COPYRIGHT, WANG LABS, INC. 1978
110 REM DATE WRITTEN = 11/15/77
120 REM MODULE FUNCTION THIS PROGRAM DOES ...
130 REM (INCLUDE FUNCTION DESC., FILES USED, SPECIAL TECHNIQUES)
140 REM A$ = NAME
150 REM B$ = ADDRESS
160 REM C = FLAG INPUT MASTER FILE
170 REM ...
180 REM ...
```

(FIGURE 2a Submodule)

```
10 REM START, 02.00, (10/04/78), THIS PROGRAM IS A PRODUCT OF WANG
LABORATORIES, INC., UNAUTHORIZED REPRODUCTION OR USE IS PROHIBITED.
(START MODULE)
20 REM SYSTEM - GRAPH UTILITY STAND-ALONE
30 REM MODULE - START
40 REM FUNCTION - INITIALIZE STORAGE AND DEVICE NUMBERS
50 REM -----
60 REM COMMON VARIABLE DESCRIPTIONS
70 REM -----
80 REM
90 REM
100 REM P7$ - PHYSICAL MEASURING UNIT (EG. CENTIMETERS)
110 REM .
120 REM
130 REM P5$(5)8 FILENAMES
140 REM P5$(1) - NAME OF PARAMETERS FILE
150 REM P5$(2) - NAME OF DATA FILE - STANDARD OR NON-STD.
160 REM P5$(3) - NAME OF LABELS FILE
170 REM P5$(4) - NAME OF CHARACTER ARRAY DATA FILE
180 REM P5$(5) - PROG. FILE FOR COMP.FUNC, READ NON-STD.
190 REM . OR 3-D
200 REM
210 REM P6$(3)3 DEVICE ASSIGNMENTS
220 REM P6$(1) - DEVICE FOR PROGRAM DISK
230 REM P6$(2) - DEVICE FOR OTHER FILES
240 REM P6$(3) - DEVICE FOR CHARACTER ARRAY FILE
250 REM ...
```

(FIGURE 2b Document Variables)

10 REM START,02.00,(10/04/78), THIS PROGRAM IS A PRODUCT OF WANG
LABORATORIES, INC., UNAUTHORIZED REPRODUCTION OR USE IS
PROHIBITED.(START MODULE).

(FIGURE 3 LINE 10 REM)

Module File Name = START
Version Number = 02.00 (Release 2 - Revision 0)
Module Date = (10/04/78)
Copyright = This program is a product of ...
Optional Description = (START MODULE)

SYSTEM 2200 RECORD DESCRIPTION FORM

The System 2200 Record Description form (Figure 5), a necessary part of application documentation, must be filled out for each type of record on a file.

Record Description Form Categories Defined

CATEGORY	DEFINITION
Application	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 design
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 a variable (i.e., A,B,C\$); in usage sequence for arrays, each variable within the array must be specified, if items in array differ (e.g., B\$(3,4), G(5))
Element Length	length of the elements; in the case 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.

* If sufficient space is not available to fully describe the field name, additional documentation should be supplied.

(FIGURE 4)

SCREEN USAGE TECHNIQUES

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.

SCREEN LAYOUT

The screen consists of 16 rows numbered 0 through 15, and 64 columns numbered 0 through 63. A larger screen is available consisting of 24 rows numbered 0 through 23 and 80 columns numbered 0 through 79.*

Input Phase

The normal procedure for the operator to start the first phase of any program (the input phase) is to key CLEAR, RETURN(EXEC), LOAD RETURN(EXEC), RUN RETURN(EXEC). The program then must clear the screen entirely. (VP/MVP may utilize the "LOAD RUN" feature).

In the input phase, line 0 of the screen is used for the program name centered on the screen. If possible, program name should remain on the screen during execution of the program. 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 program name centered ex. INVOICING MASTER FILE UPDATE;

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 Figure 6).

* 24 x 80 layout sheets are not available at this time.

** As a general rule, no prompt should take over one line. Prompts that require only a RETURN(EXEC) can exceed one line.

SCREEN DISPLAYS

First four lines of CRT during Input Phase

<i>LINE</i>	<i>DESCRIPTION</i>	<i>ACTUAL DISPLAY</i>
<i>0</i>	<i>System Messages</i>	<i>INVOICING MASTER FILE UPDATE</i>
<i>1</i>	<i>Prompt Messages</i>	<i>ENTER CUSTOMER NUMBER</i>
<i>2</i>	<i>Info requested by line 1</i>	<i>98765</i>
<i>3</i>	<i>Input errors</i>	<i>INVALID CUSTOMER NUMBER</i>

(FIGURE 6)

Execution Phase

During execution phase if no important screen display exists, the screen should be cleared from lines 1-15 and an appropriate message should be centered on line 7 describing the process taking place (i.e., "EXECUTING PROGRAM," "PRINTING REPORT," etc.). When Processing records on a 2200T, a display showing record count should appear on the screen.

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 is as follows:

Screen Only:

1. Clear lines 1-15.
2. Display screen layout as defined in specific system design.
3. When finished, clear lines 1-15 and display "END OF PROGRAM" centered on line 7 (refer to execution phase above).
4. Position cursor line 14 then "STOP."

Hardcopy Only:

1. Clear lines 1-15.
2. Line 1 is for further prompting; e.g., "IS THE PAPER POSITIONED CORRECTLY?"
3. Line 2 accepts the response.
4. When everything is verified, erase lines 1-15 of screen; Line 7 must display "PRINTING" (refer to execution phase above).
5. When finished, clear lines 1-15 and display "END OF PROGRAM" centered on line 7.

Screen and Then Hardcopy:

1. *Clear lines 1-15.*
2. *See step 2 of hardcopy only.*
3. *See step 3 of hardcopy only.*
4. *Print Screen.*
5. *Print hardcopy.*
6. *See step 5 of hardcopy only.*

```

*****
*
*      NOW LOADING REPORT GENERATOR      *
*
*****

```

(FIGURE 7 DELAY ACTIVITY DISPLAY)

GRAPH UTILITY SYSTEM 2.0

SF KEY	DESCRIPTION	SF KEY	DESCRIPTION
00	ENTER PARAMETERS	03	START PLOT
01	ENTER DATA POINTS	04	INITIALIZE SYSTEM
02	ENTER LABELS	15	CHANGE APPLICATION

(FIGURE 8 MAIN MENU FORMAT)

SYSTEM 2200 SCREEN LAYOUT

SYSTEM _____ REPORT/OPERATION _____ PAGE ____ of ____

PROGRAM NAME _____ DATE _____

MODULE _____ VERSION _____ PREPARED BY _____

1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																

FIXED CATALOG
INDEX SECTORS = 00024
END CAT. AREA = 01023
CURRENT END = 00961

NAME	TYPE	START	END	USED
GUS.215A	P	00024	00028	00005
GUS.350A	P	00029	00046	00018
STATREAD	P	00047	00053	00007
GUS.340A	P	00054	00104	00051
GUS.040A	P	00105	00176	00072
GUS.400A	P	00177	00202	00026
GUS.320D	P	00203	00207	00005
CHARF010	D	00208	00212	00005
START	P	00213	00270	00058
GUS.335A	P	00271	00304	00034
GUS.360A	P	00305	00337	00033
GUS.320E	P	00338	00347	00010
GUS.020A	P	00348	00384	00037
GUS.200A	P	00385	00485	00101
GUS.INFO	D	00486	00488	00003
GUS.310A	P	00489	00542	00054
GUS.345A	P	00543	00588	00046
GUS.320B	P	00589	00593	00005
GUS.030A	P	00594	00634	00041
GUS.300A	P	00635	00684	00050
GUS.320C	P	00685	00689	00005
GUS.000A	P	00690	00771	00082
GUS.011A	P	00772	00817	00046
GUS.321A	P	00818	00882	00065
GUS.330A	P	00883	00910	00028
GUS.MENU	P	00911	00925	00015
GUS.010A	P	00926	00956	00031
GUS.320A	P	00957	00961	00005

(FIGURE 10)

APPENDIX B STANDARDS CHECK-OFF LIST

STANDARDS CHECK-OFF LIST

- () Files are labeled and sequenced properly (START, MENUS, PROG., DATA).
- () Source does not branch to a REM statement line.
- () Start Module is a loader module only.
- () Source Code is incremented by 10.
- () Source is one statement per line.*
- () Program Variables are documented in the source code.
- () Variable Check-off list is complete.
- () Variables Q through W are reserved for ISS.
- () Record Description Form is complete.
- () DEFFN'0 through '31 are used for special function keyboard entry only.
- () Title and Release are displayed in the main menu.
- () Functions of the menus are performed with SF keys.
- () SF Keys '15 or '31 returns to the menu properly.
- () All non-related keys are locked out when prompted.
- () CRT Screen is cleared before a function change.*
- () All excessive delays display an activity message.
- () Multiple Page Reports are numbered.
- () A Top-of-form is executed prior to printing.
- () The printer is checked for selection prior to printing.
- () Special forms contain a print mask.
- () Line "10 REM" is complete (name, version, (date), copyright).
- () Source and compress file names are identical.*
- () Release Disk is compressed.*
- () Files contain extra sectors.*
- () Compressed Release Disk is free of irrelevant files.

* There is an exception to this case; consult the standards manual for more detail.

APPENDIX C DISTRIBUTION

TITLE OF MANUAL **2200 STANDARDS AND GUIDELINES**

COMMENTS:

Fold

Fold



Fold



FIRST CLASS
PERMIT NO. 16
Tewksbury, Mass.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

— POSTAGE WILL BE PAID BY —

WANG LABORATORIES, INC.
ONE INDUSTRIAL AVENUE, DEPT. 93
LOWELL, MASSACHUSETTS 01851

Attention: Quality Assurance Department

Fold

Cut along dotted line.



North America:

Alabama Birmingham Mobile	District of Columbia Washington	Louisiana Baton Rouge Metairie	New Hampshire East Derry Manchester	Oregon Beaverton Eugene	Virginia Newport News Richmond
Alaska Anchorage	Florida Jacksonville Miami Orlando Tampa	Maryland Rockville Towson	New Jersey Howell Mountainside	Pennsylvania Allentown Camp Hill Erie Philadelphia Pittsburgh Wayne	Washington Seattle Spokane
Arizona Phoenix Tucson	Georgia Atlanta	Massachusetts Boston Burlington Littleton Lowell Tewksbury Worcester	New Mexico Albuquerque	Rhode Island Cranston	Wisconsin Brookfield Madison Milwaukee
California Fresno Inglewood Los Angeles Sacramento San Diego San Francisco San Mateo Sunnyvale Tustin Ventura	Hawaii Honolulu	Michigan Grand Rapids Okemos Southfield	New York Albany Buffalo Lake Success New York City Rochester Syracuse	South Carolina Charleston Columbia	
Colorado Denver	Illinois Chicago Morton Park Ridge Rock Island	Minnesota Eden Prairie	North Carolina Charlotte Greensboro Raleigh	Tennessee Chattanooga Knoxville Memphis Nashville	Canada Wang Laboratories (Canada) Ltd. Don Mills, Ontario Calgary, Alberta Edmonton, Alberta Winnipeg, Manitoba Ottawa, Ontario Montreal, Quebec Burnaby, B.C.
Connecticut New Haven Stamford Wethersfield	Indiana Indianapolis South Bend	Missouri Creve Coeur	Ohio Cincinnati Columbus Middleburg Heights Toledo	Texas Austin Dallas Houston San Antonio	
Kansas Overland Park Wichita	Kentucky Louisville	Nebraska Omaha	Oklahoma Oklahoma City Tulsa	Utah Salt Lake City	

International Subsidiaries:

Australia Wang Computer Pty. Ltd. Sydney, NSW Melbourne, Vic. Canberra, A.C.T. Brisbane, Qld. Adelaide, S.A. Perth, W.A. Darwin, N.T.	Great Britain Wang Electronics Ltd. Northwood Hills, Middlesex Northwood, Middlesex Harrogate, Yorkshire Glasgow, Scotland Uxbridge, Middlesex	Republic of South Africa Wang Computers (South Africa) (Pty.) Ltd. Bordeaux, Transvaal Durban Capetown
Austria Wang Gesellschaft M.B.H. Vienna	Hong Kong Wang Pacific Ltd. Hong Kong	Sweden Wang Skandinaviska AB Solna Gothenburg Arloev Vasteras
Belgium Wang Europe, S.A. Brussels Erpe-Mere	Japan Wang Computer Ltd. Tokyo	Switzerland Wang S.A./A.G. Zurich Bern Pully
Brazil Wang do Brasil Computadores Ltda. Rio de Janeiro Sao Paulo	Netherlands Wang Nederland B.V. Ijsselstein	West Germany Wang Laboratories GmbH Berlin Cologne Duesseldorf Fellbach Frankfurt/M. Freiburg/Brsq. Hamburg Hannover Kassel Munich Nuernberg Stuttgart
China Wang Industrial Co., Ltd. Taipei, Taiwan	New Zealand Wang Computer Ltd. Grey Lynn, Auckland	
France Wang France S.A.R.L. Bagnolet Ecully Nantes Toulouse	Panama Wang de Panama (CPEC) S.A. Panama	
	Republic of Singapore Wang Computer Pte., Ltd. Singapore	

International Representatives:

Argentina	Kenya
Bolivia	Korea
Canary Islands	Lebanon
Chile	Liberia
Colombia	Malaysia
Costa Rica	Mexico
Cyprus	Morocco
Denmark	Nicaragua
Dominican Republic	Nigeria
Ecuador	Norway
Finland	Pakistan
Ghana	Peru
Greece	Philippines
Guatemala	Portugal
Iceland	Saudi Arabia
India	Spain
Indonesia	Sri Lanka
Iran	Syria
Ireland	Thailand
Israel	Tunisia
Italy	Turkey
Jamaica	United Arab Emirates
Japan	Venezuela
Jordan	Yugoslavia

WANG

LABORATORIES, INC.

ONE INDUSTRIAL AVENUE, LOWELL, MASSACHUSETTS 01851. TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

Printed in U.S.A.
13-2856
12-78-1M