2200 DEVELOPMENT STANDARDS AND GUIDELINES

Release 1.0

Prepared by:
Quality Assurance
January 1979
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

**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 MWP Considerations ...................................... 10
2.17 Documentation ....................................................... 11
2.18 Testing ............................................................... 12

**SECTION 3** Implementation of Standards and Guidelines .................. 13

3.1 Narrative .............................................................. 13
3.2 Development Flow .................................................... 13

**SECTION 4** Standard/Guideline Request ................................... 14

4.1 Overview .............................................................. 14
4.2 Request Form ........................................................ 15

**APPENDIX A** Examples ..................................................... 16

**APPENDIX B** Standards Check-off List .................................... 17

**APPENDIX C** Distribution .................................................. 20

Page

1

2

3

4

6

6

7

7

7

7

8

8

9

9

9

9

9

10

11

12

13

14

15

16

17

20

29

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, respectively (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:
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#O, (e.g., 300 LOAD DC T #0, "PAYROI0A," 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(0E); 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
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.
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

16
Wang BASIC Variable Check-off List

<table>
<thead>
<tr>
<th>PROGRAM NAME</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERSION</td>
<td>PROGRAMMER</td>
</tr>
<tr>
<td>SYSTEM</td>
<td></td>
</tr>
</tbody>
</table>

| NUMERIC SCALARS |
| FORMAT = MN |

| NUMERIC ARRAYS |
| FORMAT = MN( |

| ALPHA NUMERIC SCALARS |
| FORMAT = MNS |

| ALPHA NUMERIC ARRAYS |
| FORMAT = MNS( |

NOTE:
0 = NON COMMON
1 = COMMON DEFINED BY THIS MODULE
2 = COMMON DEFINED BY PREVIOUS 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 P7$  - PHYSICAL MEASURING UNIT (EG. CENTIMETERS)
90 REM .
100 REM P5$(5)8 FILENAMES
120 REM ...  P5$(1) - NAME OF PARAMETERS FILE
140 REM P5$(2) - NAME OF DATA FILE - STANDARD OR NON-STD.
160 REM P5$(3) - NAME OF LABELS FILE
180 REM P5$(4) - NAME OF CHARACTER ARRAY DATA FILE
190 REM P5$(5) - PROG. FILE FOR COMP. FUNC., READ NON-STD.
200 REM OR 3-D
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)
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

<table>
<thead>
<tr>
<th>CATEGORY</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>same name as application program</td>
</tr>
<tr>
<td>Application ID</td>
<td>first 4 characters of program name; e.g., PAY1</td>
</tr>
<tr>
<td>File Name</td>
<td>8 characters, detailed in Section 3.3 (FILE NAME)</td>
</tr>
<tr>
<td>Date</td>
<td>when record design</td>
</tr>
<tr>
<td>Record Name</td>
<td>short English description</td>
</tr>
<tr>
<td>Record ID Code</td>
<td>indicates record type as in power company example, Boston Edison was Al</td>
</tr>
<tr>
<td>Field Name</td>
<td>English description of the field</td>
</tr>
<tr>
<td>Type</td>
<td>Alpha or numeric field</td>
</tr>
<tr>
<td>Variable</td>
<td>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))</td>
</tr>
<tr>
<td>Element Length</td>
<td>length of the elements; in the case arrays, the length of the element within the array</td>
</tr>
<tr>
<td>Tape Length</td>
<td>for numerics, always 9 bytes; for alphas, always the length of the field plus one</td>
</tr>
<tr>
<td>Key</td>
<td>place a check ( ) if to be used as a key field</td>
</tr>
<tr>
<td>Notes*</td>
<td>special remarks such as parameters of the field, how it relates to other fields, or purpose of the fields.</td>
</tr>
</tbody>
</table>

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

(FIGURE 4)
<table>
<thead>
<tr>
<th>FIELD NAME</th>
<th>TYPE</th>
<th>VAR.</th>
<th>ELEM LEN.</th>
<th>TAPE/ DISK LEN</th>
<th>KEY</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD ID CODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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:

1. The operator always must see the same general layout on the screen, regardless of what program is being run.
2. 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

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

(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:

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:

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:

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.
GRAPH UTILITY SYSTEM 2.0

<table>
<thead>
<tr>
<th>SF KEY</th>
<th>DESCRIPTION</th>
<th>SF KEY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>ENTER PARAMETERS</td>
<td>03</td>
<td>START PLOT</td>
</tr>
<tr>
<td>01</td>
<td>ENTER DATA POINTS</td>
<td>04</td>
<td>INITIALIZE SYSTEM</td>
</tr>
<tr>
<td>02</td>
<td>ENTER LABELS</td>
<td>15</td>
<td>CHANGE APPLICATION</td>
</tr>
</tbody>
</table>

(Figure 7 DELAY ACTIVITY DISPLAY)

(Figure 8 MAIN MENU FORMAT)
<table>
<thead>
<tr>
<th>MODULE</th>
<th>VERSION</th>
<th>PREPARED BY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**FIXED CATALOG**

**INDEX SECTORS = 00024**

**END CAT. AREA = 01023**

**CURRENT END = 00961**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>START</th>
<th>END</th>
<th>USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUS.215A</td>
<td>P</td>
<td>00024</td>
<td>00028</td>
<td>00005</td>
</tr>
<tr>
<td>GUS.350A</td>
<td>P</td>
<td>00029</td>
<td>00046</td>
<td>00018</td>
</tr>
<tr>
<td>STATREAD</td>
<td>P</td>
<td>00047</td>
<td>00053</td>
<td>00007</td>
</tr>
<tr>
<td>GUS.340A</td>
<td>P</td>
<td>00054</td>
<td>00104</td>
<td>00051</td>
</tr>
<tr>
<td>GUS.040A</td>
<td>P</td>
<td>00105</td>
<td>00176</td>
<td>00072</td>
</tr>
<tr>
<td>GUS.400A</td>
<td>P</td>
<td>00177</td>
<td>00202</td>
<td>00026</td>
</tr>
<tr>
<td>GUS.320D</td>
<td>P</td>
<td>00203</td>
<td>00207</td>
<td>00005</td>
</tr>
<tr>
<td>CHARF010</td>
<td>D</td>
<td>00208</td>
<td>00212</td>
<td>00005</td>
</tr>
<tr>
<td>START</td>
<td>P</td>
<td>00213</td>
<td>00270</td>
<td>00058</td>
</tr>
<tr>
<td>GUS.335A</td>
<td>P</td>
<td>00271</td>
<td>00304</td>
<td>00034</td>
</tr>
<tr>
<td>GUS.360A</td>
<td>P</td>
<td>00305</td>
<td>00337</td>
<td>00033</td>
</tr>
<tr>
<td>GUS.320E</td>
<td>P</td>
<td>00338</td>
<td>00347</td>
<td>00100</td>
</tr>
<tr>
<td>GUS.020A</td>
<td>P</td>
<td>00348</td>
<td>00384</td>
<td>00037</td>
</tr>
<tr>
<td>GUS.200A</td>
<td>P</td>
<td>00385</td>
<td>00485</td>
<td>00101</td>
</tr>
<tr>
<td>GUS.INFO</td>
<td>D</td>
<td>00486</td>
<td>00488</td>
<td>00003</td>
</tr>
<tr>
<td>GUS.310A</td>
<td>P</td>
<td>00489</td>
<td>00542</td>
<td>00054</td>
</tr>
<tr>
<td>GUS.345A</td>
<td>P</td>
<td>00543</td>
<td>00588</td>
<td>00046</td>
</tr>
<tr>
<td>GUS.320B</td>
<td>P</td>
<td>00589</td>
<td>00593</td>
<td>00005</td>
</tr>
<tr>
<td>GUS.030A</td>
<td>P</td>
<td>00594</td>
<td>00634</td>
<td>00041</td>
</tr>
<tr>
<td>GUS.300A</td>
<td>P</td>
<td>00635</td>
<td>00684</td>
<td>00050</td>
</tr>
<tr>
<td>GUS.320C</td>
<td>P</td>
<td>00685</td>
<td>00689</td>
<td>00005</td>
</tr>
<tr>
<td>GUS.000A</td>
<td>P</td>
<td>00690</td>
<td>00771</td>
<td>00082</td>
</tr>
<tr>
<td>GUS.011A</td>
<td>P</td>
<td>00772</td>
<td>00817</td>
<td>00046</td>
</tr>
<tr>
<td>GUS.321A</td>
<td>P</td>
<td>00818</td>
<td>00882</td>
<td>00065</td>
</tr>
<tr>
<td>GUS.330A</td>
<td>P</td>
<td>00883</td>
<td>00910</td>
<td>00028</td>
</tr>
<tr>
<td>GUS.MENU</td>
<td>P</td>
<td>00911</td>
<td>00925</td>
<td>00015</td>
</tr>
<tr>
<td>GUS.010A</td>
<td>P</td>
<td>00926</td>
<td>00956</td>
<td>00031</td>
</tr>
<tr>
<td>GUS.320A</td>
<td>P</td>
<td>00957</td>
<td>00961</td>
<td>00005</td>
</tr>
</tbody>
</table>

*(FIGURE 10)*
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
North America:

Alabama
Birmingham
Mobile

Alaska
Anchorage

Arizona
Phoenix
Tucson

California
Fresno
Inglewood
Los Angeles
Sacramento
San Diego
San Francisco
San Mateo
Sunnyvale
Tustin
Ventura

Colorado
Denver

Connecticut
New Haven
Stamford
Wethersfield

District of Columbia
Washington

Florida
Jacksonville
Miami
Orlando
Tampa

Georgia
Atlanta

Hawaii
Honolulu

Illinois
Chicago
Morton
Park Ridge
Rock Island

Indiana
Indianapolis
South Bend

Kansas
Overland Park
Wichita

Kentucky
Louisville

Louisiana
Baton Rouge
Metairie

Maryland
Rockville
Towson

Massachusetts
Boston
Cambridge
Littleton
Lowell
Tewksbury
Worcester

Michigan
Grand Rapids
Okemos
Southfield

Minnesota
Eden Prairie

Missouri
Creve Coeur

Nebraska
Omaha

Nevada
Reno

New Hampshire
East Derry
Manchester

New Jersey
Howell
Mountainside

New Mexico
Albuquerque

New York
Albany
Buffalo
Lake Success
New York City
Rochester
Syracuse

North Carolina
Charlotte
Greensboro
Raleigh

Ohio
Cincinnati
Columbus
Middleburg Heights
Toledo

Oklahoma
Oklahoma City
Tulsa

Oregon
Beaverton
Eugene

Pennsylvania
Allentown
Camp Hill
Erin
Philadelphia
Pittsburgh
Wayne

Rhode Island
Cranston

South Carolina
Charleston
Columbia

Tennessee
Chattanooga
Knoxville
Memphis
Nashville

Texas
Arlington
Dallas
Houston
San Antonio

Utah
Salt Lake City

Virginia
Newport News
Richmond

Washington
Seattle
Spokane

Wisconsin
Brookfield
Madison
Milwaukee

Canada
Wang Laboratories
(Canada) Ltd.
Don Mills, Ontario
Calgary, Alberta
Edmonton, Alberta
Winnipeg, Manitoba
Ottawa, Ontario
Montreal, Quebec
Burnaby, B.C.

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.

Austria
Wang Gesellschaft M.B.H.
Vienna

Belgium
Wang Europe, S.A.
Brussels

Brazil
Wang do Brasil
Compuadores Ltda.
Rio de Janeiro
Sao Paulo

China
Wang Industrial Co., Ltd.
Taipei, Taiwan

France
Wang France S.A.R.L.
Bagnoulet

Georgia
Atlanta

Great Britain
Wang Electronics Ltd.
Northwood Hills, Middlesex
Northwood, Middlesex
Harrogate, Yorkshire
Glascow, Scotland
Uxbridge, Middlesex

Hong Kong
Wang Pacific Ltd.
Hong Kong

Japan
Wang Computer Ltd.
Tokyo

Netherlands
Wang Nederland B.V.

New Zealand
Wang Computer Ltd.

Panama
Wang de Panama

Republic of Singapore
Wang Computer Pte. Ltd.

Republic of South Africa
Wang Computer
(South Africa) Pty. Ltd.

Switzerland
Wang Skandinaviska AB

Sweden
Wang Skandinaviska AB


International Representatives:

Argentina
Bolivia
Canary Islands
Chile
Colombia
Costa Rica
Cyprus
Denmark
Dominican Republic
Ecuador
Finland
Ghana
Greece
Guatemala
Iceland
India
Indonesia
Iran
Ireland
Israel
Italy
Jamaica
Japan
Jordan

Kenya
Korea
Lebanon
Libya
Malaysia
Mexico
Morocco
Nicaragua
Nigeria
Norway
Pakistan
Peru
Philippines
Portugal
Saudi Arabia
Spain
Sri Lanka
Syria
Thailand
Turkey
United Arab Emirates
Venezuela
Yugoslavia

LABORATORIES, INC.
ONE INDUSTRIAL AVENUE, LOWELL, MASSACHUSETTS 01851. TEL: (617) 851-4111. TWX 710 343-8679. TELEX 94-7421