Wang Micro-VP Tool Box Option Software

The "Wang Tool Box Option Software" is a loose-leaf collection of Wang 2200 Basic-2 programs that were developed to aid and assist in the development of Wang 2200 systems. The programs may be accessed from a simple command menu.

The 2200 Technical Support group has used these programs and thought they might be useful to the field as well.

Programs are Unsupported

Programs are a Loose-leaf collection subject to change

Documentation is Reset, LOAD, RUN, RETURN
and read the prompts on the CRT

The "Wang Tool Box Option Software" diskette includes:

Utilities and Program Development Aids
eg.
Cross-reference and Sort catalog area

Clock and calendar

Diagnostic Aids

Monitor Partition status

Prepare or list TC format data

Menu entry to system @MENU

An installation program to move files to a system platter

Wang micro-VP Tool Box Option Software

Index:

Wang Micro-VP Tool Box Option Software

The "Wang 2200 Tool Box Option Software" is a loose-leaf collection of Wang 2200 Basic-2 programs that were developed to aid and assist in the development of Wang 2200 systems. The programs may be accessed from a simple command menu.

```
Tool Box Option                           MM/DD/YY           What do you wish to do?
TO OPERATE -- Press S.F. KEY or DIGIT corresponding to name,
or position # via Alpha, RETURN, SPACE or BACKSPACE and key RUN

  #'  1 to Tool Box Utilities and program development aids
   '  2 to Clock and calendar
   '  3 to Tool Box Diagnostic Aids
   '  4 to LOAD RUN xx from another surface
   '  5 to Monitor Partition status
   '  6 to Prepare or list TC format data
   '  7 to @MENU interaction
```

```
ToolBox Option Utilities                  MM/DD/YY           What do you wish to do?
TO OPERATE -- Press S.F. KEY or DIGIT corresponding to name,
or position # via Alpha, RETURN, SPACE or BACKSPACE and key RUN

  #'  1 to Sort the disk catalog area
   '  2 to Cross-reference listing
   '  3 to Program or system compare
   '  4 to @MOVEFIL disk to disk file copy
   '  5 to File Z A P
   '  6 to Analyze data file structure
   '  7 to Disk catalog examination
   '  8 to Search Programs for verbs
   '  9 to Flow chart maker
   ' 10 to Map disk for program-call integrity
   ' 11 to Data file comparison
   ' 12 to Notes on Tool Box utilities
   ' 13 to Number conversions
```

```
Tool Box Diagnostics                      MM/DD/YY           What do you wish to do?
TO OPERATE -- Press S.F. KEY or DIGIT corresponding to name,
or position # via Alpha, RETURN, SPACE or BACKSPACE and key RUN

  #'  1 to What disks are on the system?
   '  2 to Scan disks on the system
   '  3 to CRT character set and keyboard values
   '  4 to Show status of Printers, TC boards, & Terminal
   '  5 to Examine ASKACALL file
   '  6 to Analyze $GIO statements
   '  5 to Analyze disk index
```

Installation

The "Tool Box Option" diskette is provided on a single surface. An installation program "TBO.INST" may be used to move the files to a system platter. The entry program is "START" which calls "STARTTBO".

The menus activated by the utility "STARTTBO" operate off a list of DATA statements which normally overlay the code from 9000 - end. These DATA statements are of the form:

```
 Bytes 1-8 Program module name
 Byte  9 value = space if loading a program
               = ]  if loading a menu overlay (new code for statements 9000-end).
               = c  if saving COMmon variables, eg. disk addresses.
               = @  if loading the standard @MENU program.
 Bytes 10-70 are descriptive text to appear on menu line.
```

Line 15 of module "TBO UTIL" allows you to specify the most common disk surfaces on your system. When asked to specify a specific surface the operator may then key a single digit or a full disk address and EXEC.

```
    0015 COM D1$(12)4: D1$()="310 B10 320 B20":REM/.up to 9 disks

9000 REM   ....!...V1....!....2....!....3....!....4....!....5....!....6....
9... DATA "Tool Box Option"
9... DATA "TBO UTIL]TBO Utilities
9... DATA "@CLOCK   Clock and calendar
9... DATA "TBO DIAG]Diagnostic Aids
9... DATA "TBO TCPL]Prepare or list TC format data
9... DATA "@SYSMVPB@@MENU
9... DATA " "
```

Within the menu "TBO UTIL"

```
9000 REM TBO UTIL
9... DATA "Tool Box Option Utilities"
9... DATA "TBO.SDCOcSort the disk catalog area
9... DATA "TBO.CRFOcCross-reference listing
9... DATA "TBO.CMPScProgram or system comparison
9... DATA "@MOVEFIL @MOVEFIL
9... DATA "TBO.ZAP  File Zap
9... DATA "TBO.ANDFcAnalyze data file structure
9... DATA "TBO.XDC cDisk catalog examination
9... DATA "TBO.SPV cSearch programs for verbs
9... DATA "TBO.FLOWcFlow chart maker
9... DATA "TBO.DMAPcMap disk for program call integrity
9... DATA "TBO.CMD cData file comparison
9... DATA "TBO NOTE Notes on TBO utilities
9... DATA "TBO NUMC]Number conversions
9... DATA " "
9... DATA "TBO.CRF1 Crossref COMmon
9... DATA "TBO.CRF2 Crossref set-up
9... DATA "TBO.CRF3 Crossref mainline
9... DATA "TBO.SDC1 SORTCAT  mainline
9... DATA " "
```

Within the menu "TBO DIAG"

```
9000 REM TBO DIAG
9... DATA "Tool Box Diagnostics"
9... DATA "TBO.XDAD What disks are on the system?
9... DATA "TBO.CRT  CRT character set and keyboard values
9... DATA "TBO.STAT Show status of Printers, TC boards, & terminal
9... DATA "TBO.XASK Examine ASKACALL file
9... DATA "TBO.$GIO Analyze $GIO statements
9... DATA "TBO.ADI  Analyze disk index
9... DATA " "
```

II. Abstracts for Tool Box Option Utilities

Function: Sort the disk catalog area
Modules: TBO.SDCO - TBO.SDC1

Abstract: Sorts the entries of the disk catalog into an ordered sequence. The listing may be in name sequence or in sector number, or reverse sector number order. Program or data file names are listed item by item in separate groups. Listings requested by name may include all or a subset of the files stored.
BASIC-2 command LIST S DCT lists all file names in hash sequence order.

Function: Cross-reference listing
Modules: TBO.CRFO - TBO.CRF1 - TBO.CRF2 - TBO.CRF3
Abstract: One or more program files on any system disk may be listed on numbered and titled pages. For each program requested the listing consists of decompressed program statements followed by a cross-referenced listing of variables used, special function, and numbered statement references.

BASIC-2 commands operate on a program in the user partition:

LIST	lists all lines.
LISTSD	lists a Section Decompressed.
LIST V	lists Variables used.
LIST '	lists Special Function references.
LIST #	lists numbered statement references.

Function: Program or system comparison
Modules: TBO.CMPS
Abstract: Program files on any system disks are compared line by line. Differences between the two files are displayed line by line. Two system disks may be compared by activating a list of DATA statement names.

BASIC-2 commands -- None.

Function: @MOVEFIL disk to disk file copy
Modules: @MOVEFIL
Abstract: This program from the standard BASIC-2 release copies program and data files from one disk surface to another.

Function: File Z A P
Modules: TBO.ZAP
Abstract: A single disk surface may be examined. The display shows the contents of a single sector in hexidecimal notation and in ASCII.

1. Items in the catalog index area are flagged as AP (Active Programs), AData (Active Data).
2. Data files are displayed with highlighted attribute bytes.
3. Program files are displayed with highlighted RETURN codes.

Caution, you have the ability to change any all data in a selected sector.
BASIC-2 commands -- None.

Function: Analyze data file structure

Modules: TBO.ANDF

Abstract: This program analyzes a catalogued data file for structure. The output is a summary of the SAVE statements used to create the file.

```
Example  File name=@SYSFILE
         File size = 32 Sectors.
         File=@SYSFILE   BASIC-2 data structure -- SAVE number 1
         $8     Scalar   $32   $( 2)16  $(16)8  $(33)3   $(15)13

Example  File name=BSC*010A
         File size = 40 Sectors.
         File=BSC*010A   BASIC-2 data structure -- SAVE number 1
         $( 256)16
         File=BSC*010A   BASIC-2 data structure -- SAVE number 2
         $( 256)16
         END OF FILE
```

Function: Disk catalog examination

Modules: TBO.XDC

Abstract: This program lets the operator 1) examine the disk catalog, 2), Examine a disk, or 3). Search disk for programs. I use 3) to locate programs by sector number when a disk index has been clobbered.

Function: Search programs for verbs

Modules: TBO.ANDF

Abstract: This program lets you search a list of programs for specific verbs. It was originally written to determine which kind of system was required to run a certain program. I have used it in two different environments:

1) to search systems for COM statements when trying to reduce memory requirements,
2) for places where string variables are set to quotable values, (useful when translating English menues to Spanish).

Function: Flow chart maker

Modules: TBO.FLOW

Abstract: This program was an early attempt to make flow charts from BASIC-2 code. It draws boxes around BASIC-2 code. Who cares? Someone who is required to submit flow diagrams of their program. The cross-reference program when used with REM% comments and variable annotation is far superior.

Function: Map disk for program call integrity

Modules: TBO.DMAP

Abstract: This program traces the program map of a disk by keying off the program names and descriptions in load modules. In a TBO menu structured disk it will show which menu picks are further menues, which are programs, and which dead-ended", ie. programs that are non-existent on the disk.

Function: Data File comparison
Modules: TBO.CMD
Abstract: Data files on any system disks are compared byte by byte. Differences between the two files are highlighted and displayed sector by sector.

BASIC-2 commands -- None.

Function: Partition status
Modules: TBOPSTAT
Abstract: This program built from the SYSTEM UTILITY "@PSTAT" also shows the Device Table on one line toward the bottom of the screen. This addition is useful to determine if some device might be used or hogged by another partition.

Function: Disk status
Modules: TBO.XDAD
Abstract: This program displays the status of all disks on the system. It is a static display of possible disk addresses on drives 10, 20. and 30 It shows Cur.end -Max.,Access errors, disk not configured, disk is unavailable.

```
 THIS IS THE STATUS OF POSSIBLE DISKS ON THE SYSTEM
Disk Controller 10  Disk type CO  ...   2270 type
   D10 is the equivalent of B10     !   D11 is the equivalent of 310
 1 D10 Yes Cur.END=17414 Max.=38911 ! 2 D11.. etc.
 3 350 Access error = 98
```

Function: Scan disks on the system
Modules: TBO.DSCN

Abstract: This program displays file names from the disk catalog index.
BASIC-2 commands -- LIST DCT W

Function: CRT character set and keyboard values
Modules: TBO.CRT
Abstract: This program displays the full CRT character set. After the initial display, various keys can be depressed to determine the hex and ASCII values of keystrokes. (S.F.=hh) shows special function keys, ie those with an [illegible] level.

Function: T.C. Board status
Modules: TBO.STAT
Abstract: This program displays the status of peripherals on selected device addresses. Printers are tested on 215, 216, 217, and 218. T.C. boards are tested on 6 addresses from 1A to 1F. For T.C. boards the program shows Ready/Not, B,C, or D, and memory available. The status of the operating system and user terminal configuration are also displayed.

BASIC-2 command LIST DT shows addresses configured, selected or hogged.

Function: Examine ASKACALL file
Modules: TBO.XASK
Abstract: This program displays the contents of the configuration file "ASKACALL" used in the ASC, BSC, and 2200/3270 emulations.
N$ is the name given the emulation by the operator,
W$ in non-3270 displays the modules loaded.
Z$ bytes 01-20 in hex -- are parameters for the microcode.
bytes 21-64 in ASCII are responses to prompts.

Function: Analyze $GIO arguments
Modules: TBO.$GIO
Abstract: This program treats the variable D$() as a $GIO microcommand. It will do a disassembly on the array and create a screen display. The content is a listing from chapter 15 of the BASIC-2 programmers manual in relation to each microcommand found.

Function: Analyze disk index
Modules: TBO.ADI
Abstract: This program analyzes the disk index for efficiency.

Tool Box Option Utility

III. Operating Instructions for selected utilities

Tool Box Option Utility -- Sort the disk catalog area. Page 1 of 2.

Abstract: Sorts the entries of the disk catalog into an ordered sequence. The listing may be in name sequence or in sector number, or reverse sector number order. Program or data file names are listed item by item in separate groups. Listings requested by name may include all or a subset of the files stored.

Modules: TBO.SDC0 - TBO.SDC1
Equipment used: CRT / keyboard and optional printer.

Operating instructions:

```
Display  REQUEST NUMBER= 1                          SORT DISK FILE CATALOG NAMES
          1 =310    2 =B10    3 =320    4 =B20
Prompt 1 Disk surface   Key single digit or any valid disk address 1__
Respond  EXEC
   or    digit and EXEC
   or    disk address and EXEC
Display   1 =310

Prompt 2 Sort by  0=Name  1=Sector  2=-Sector                (DEFAULT)=0 ? _
Respond  EXEC             for sequence by Name
   or    0 and EXEC       for sequence by Name
   or    1 and EXEC       for sequence in Sector number order
   or    2 and EXEC       for sequence in -Sector order, ie. last names entered

Prompt 3 Common Root ID ? _                 Asked if Prompt 2 response was 0.
Respond  EXEC             to collect names of all files
         1-8 characters and EXEC to collect by common ID.  A virgule ("/") in
         any position may be used for masked searching.

Display  TYPE OF FILE NAMES TO SORT
         PROGS+DATA     A          S           ALL
         PROGRAMS       AP         SP          P=AP+SP
         DATA           AD         SD          D=AD+SD
Prompt 4 Category     Active     Scratched                   (DEFAULT)=AP? _
Respond  EXEC             for sequenced list of Active Program file names.
   or    AP and EXEC      for Active Programs.
   or    SP and EXEC      for Scratched Programs.
   or    AD and EXEC      for Active Data file names.
   or    SD and EXEC      for Scratched Data file names.
   or    A  and EXEC      for Active Program and Data file names.
   or    S  and EXEC      for Scratched Programs and Data file names.
   or    ALL and EXEC     for sequenced list of AP, AD, SP, and SD file names.
   or    DATA and EXEC    for sequenced list of AP, AD, SP, and SD file names
                          contained within a list of DATA statements overlaid
                          over lines 9000+.  The output list will show files
                          contained on the surface and also denote those missing.

Prompt 5 Output to:       0=CRT  1=215  2=204  3=216         (DEFAULT)= 0 ? _
Respond  EXEC or digit and EXEC.

Prompt 6 List wanted      0=Cat.data  1=plus REMS            (DEFAULT)= 1 ? _
         Note "plus REMS" means to display a catalog index line and:
              for programs a portion of the first line of a program file if
              coded as a REM or % (image statement).
              for data files a description for the data file name if found in
              a list of DATA statements overlaid over lines 9000+.
Respond  EXEC or digit and EXEC.
```

Prompt 7 INPUT NON-ZERO FOR MORE DATA?
Respond EXEC to start Sort collection
or key and EXEC to prompt for an additional sort collection..

- -

Additional options:

These options may be invoked after the collection phase of the sort.

'1-Cat.only --
displays for each file only the single line of catalog information.

'2-Cat.with REMs --
displays for each file the catalog line plus a REM. (see prompt 6 elaboration).

'7=BEGIN --
starts the display over again with the first item.

'14 date + --
prompts for a date mm/dd/yy and then searches for program files entered that date or after. The program must have an initial REM or % statement with a date included. The included date may be of mm/dd/yy or yy/mm/dd format.

'15 RECALL ID
Skip through the list of collected names and begin with a matching ID.

- -

Output:

```
Screen   DISK CATALOGUE SORTED BY NAME
                   INDEX SECTORS =      nn
                   END CAT. AREA =   aaaaa
                   CURRENT END   =   aaaaa
         SEARCHING FOR AP                [Standard index/New index method
         CAT. SECTOR= s    FOUND ITEM = nn filename
                   AP ITEMS FOUND=      ccc
              ITEM   NAME      TYPE     START     END     USED     FREE    +USED

                 1   TBO DIAG   P       S.s.#   E.s.#       6        1        6
                     text of REM or % statement if first line of program.

                 2   TBO UTIL   P       S.s.#   E.s.#       6        2       12
                     text of REM or % statement if first line of program.

         '1-Cat.only  '2-Cat.with REMs  '7=BEGIN  '14 date +  '15 RECALL ID

Printer  DISK CATALOGUE SORTED BY NAME
                   INDEX SECTORS =      nn
                   END CAT. AREA =   aaaaa
                   CURRENT END   =   aaaaa

                   AP ITEMS FOUND=      ccc
              ITEM   NAME      TYPE     START     END     USED     FREE    +USED

                 1   TBO DIAG   P       S.s.#   E.s.#       6        1        6
                     text of REM or % statement if first line of program.

                 2   TBO UTIL   P       S.s.#   E.s.#       6        2       12
                     text of REM or % statement if first line of program.
```

- -

Abstract: This program displays a clock and calendar on a 2200 terminal. The clock and calendar can be used for reminder messages for the day. If a universal global area is set reminders may be sent from terminal to terminal.

Modules: @CLOCK -- DATETIME -- TBO.MSGS

Equipment used: CRT / keyboard and MXE controller.

Operating instructions:

for "DATETIME"
Display Enter Date and Time
MM/DD/YY HH:MM

Respond Valid date and time for the MXE controller.

for "@CLOCK" Display

```
                         FRIDAY   JANUARY 24, 1986

                      W A N G   2 2 0 0   M i c r o - V P

   ***      ******       ******     ******         ******    ******
    **     **    **     **    **   **    **     **    **  **    **
    **           **  +  **    **   **    **  +  **    **  **    **
    **         **       **    **   **    **     **    **  **    **
    **       **      +  **    **   **    **  +  **    **  **    **
    **     **           **    **   **    **     **    **  **    **
   *****   ********      ******     ******         ******    ******


          JANUARY 1986                         FEBRUARY 1986
   SUN MON TUE WED THU FRI SAT          SUN MON TUE WED THU FRI SAT
                 1   2   3   4                                    1
     5   6   7   8   9  10  11            2   3   4   5   6   7   8
    12  13  14  15  16  17  18            9  10  11  12  13  14  15
    19  20  21  22  23  24  25           16  17  18  19  20  21  22
    26  27  28  29  20  31               23  24  25  26  27  28
```

- - - - - - - - - - - - - - - - - -

Respond space to find instructions
Key NOTE (s.f. '6) to display messages for this terminal
-- permanent reminders are DATA statements at lines 6100+
Key INSERT to enter current reminder messages
Key DELETE to remove messages
Key <- PREV or NEXT -> to shift months display
Key CANCEL/EDIT to return to main menu
Key EXEC to show current month

Respond CANCEL/EDIT to exit to main module saving outstanding reminders.

NOTE: Reminder messages to display in BOLD TYPE and standard type may be inserted to occur at set times. Nine characters are displayed BOLD TYPE.

- - - - - - - - - - - - - - - - - -

NOTE: Holidays and weekends are displayed in highlighted reverse intensity. The current day is displayed in boxed in high-intensity blinking.

Holidays are coded in A$="mm/dd, ... " form on line 125.

Additional instructions for @CLOCK.

```
        hh:mm START.JOB FileName ---Special NOTE form to LOAD RUN "FileName"
        Key '31 to send a message to another terminal.

respond '10 to INSERT messages to your terminal's Clock file.
display
         hh:mm BOLD TYPE insert message
        hh:mm text for message 1
        hh:mm text for message 2
respond hh:mm text for new message
 or     mm/dd hh:mm text for new message

respond '9  to DELETE messages from your terminal's Clock file.
display
         hh:mm or ALL -- DELETE message
        hh:mm text for message 1
        hh:mm text for message 2
respond hh:mm  and RETURN to delete one of today's messages
 or     mm/dd hh:mm and RETURN.
 or     ALL and RETURN
```

Enhancements for an improved clock functionality.

1. File "SYS CRTS" may be added to describe terminals on your system.
NOTE descriptions for system terminals are in a file "SYS CRTS"

```
8000 REM .SYS CRTS
   :REM. "Msg.File # CRT description
8010 DATA "@CLOCKM" 1  Main system terminal
8020 ...  "MSGS.TBO 2  Main system terminal
```

2. A global message area may be established for inter-terminal communications.
Line 24 of @CLOCK looks for this partition.

```
0024 M0=1
     : SELECT @PART "3270UNIV"
     : ERROR M0=0
```

3. Use a file called "SYS COM" as the first Universal COMmon memory module.

```
0010 REM SYS COM mm/dd/yy COMmon memory for this system
0015 REM .Called from WAITDATE
0020 COM @M$(8)50
     : REM /.# Terminals for CLOCK messages.
0070 LOAD T "3270U2"        Calls next Universal partition module.
```

Tool Box Option Utility -- Disk status.

Function: What disks are on the system?
Modules: TBO.XDAD
Abstract: This program displays the status of all disks on the system. It is a static display of possible disk addresses on drives 10, 20. and 30 It shows Cur.end -Max.,Access errors, disk not configured, disk is unavailable.

```
 THIS IS THE STATUS OF POSSIBLE DISKS ON THE SYSTEM
Disk Controller 10  Disk type C0  ...   2270 type
   D10 is the equivalent of B10     !   D11 is the equivalent of 310
 1 D10 Yes Cur.END=17414 Max.=38911 ! 2 D11.. etc.
 3 350 Access error = 98

Disk controller 20  Disk type D0  ...   DS cabinet


THIS IS THE STATUS OF POSSIBLE DISK ADDRESSING ON THIS SYSTEM
Key '0 to EXIT   '1 to TBO.STAT    '15 RECALL  - other for disk notes

(canned informational display)
2270     -- Dual or triple   (1231)  maximum sectors, white label
2260    ---   two master and two slave addresses
2280    ---   six master and six slave addresses
Winchester                   (3873)  maximum sectors, red label
2275's   -- DSDD max sectors (1292)  maximum sectors, 5 1/4 inch
2275-30 -- DSDD on D.0       (1292)    Winchester on D.1 and D.2 (18900)
2275-60 -- No diskette   -- 4 Winchester surfaces on D.0 thru D.3 (65023)
Phoenix -- No diskette   --             surfaces on D.0 thru D.F (52763)

DS cabinet DSDD on D0 (1292)            surface  on D.0
           DSHD on D0 (4159)            surface  on D.0
   Sw.1    Winchester drives            surfaces D.1 thru D.E (40 bit = 0)
           DS-10R                       surface  D.F
   Sw.2    Winchester drives            surfaces D.1 thru D.E (40 bit = 1)


D.0
```

Tool Box Option Utility -- T.C. Board status

Function: Show status of Printers, T.C. boards, and Terminal

Abstract: This program displays the status of peripherals on selected device addresses. Printers are tested on 215, 216, 217, and 218. T.C. boards are tested on 6 addresses from 1A to 1F. The status of the operating system and user terminal are also displayed.

Modules: TBO.STAT

Screen display

```
                                  2200 Device status


Terminal status
!*2436DE R0101 19200B 8+0 (USA) !

Terminal number       1            ! CPU type          MVP         !
Partition number      9            ! MVP Release       2.6         !
Partition memory     56            ! CRT size           80         !

Printers
215 Unavailable                    ! 216 Not Ready                 !
217 Unavailable                    ! 218 Unavailable               !

T.C. Boards
01A Unavailable                    ! 01B Unavailable               !
01C B. 32K                         ! 01D D. 64K                    !
01E B.  8K                         ! 01F Unavailable               !

Key '0 to START  '1 to What disks?   other key to retest here
```

Tool Box Option Utility -- LOAD RUN xx from another surface

Function: LOAD RUN xx from another surface

Abstract:

The program does a SELECT #1 with the given disk address. It then tries a LIMITS T #1, with the file name given to assure the file name given is a valid program on the stated disk surface.

If false, reprompting occurs. S.F. '0 may be used to reload "START".

If true, TBO.LRUN does a SELECT #0 with the new disk surface; the new program is activated with a LOAD RUN command.

Modules: TBO.LRUN

Display Activate a system from another platter

Prompt 1 On disk surface ___
Respond hhh and EXEC where hhh is a valid disk address.

Prompt 2 Program START___
Respond EXEC to load "START".
else file name and EXEC.

Function: Cross-reference listing

Abstract: One or more program files on any system disk may be listed on numbered and titled pages. For each program requested the listing consists of decompressed program statements followed by a cross-referenced listing of variables used, special function, and numbered statement references.

Modules: TBO.CRF0 - TBO.CRF1 - TBO.CRF2 - TBO.CRF3

Equipment used: CRT / keyboard, disk and optional printer.

Operating instructions:

Display TBO xref -- System code = MVP with 56 K memory
CROSSRF -- S.F. Entry points
From prompt module:
'0 Change Disk media
'1 Pick-up prompting in program names
'5 display verb atoms '15 Show S.F. actions
'12 writeup '14 List options
From cross-reference listing mainline:
'10 Summary to Printer/CONTINUE '11 Summary only

Prompt 1 Output to: 0=CRT 1=215 2=204 3=216 (DEFAULT)= 0 ? _
Respond EXEC or digit and EXEC.

Prompt 2 DATE?
Respond any character stream (no commas) will be part of a large print title put at the top of each printed page.
EXEC or digit and EXEC.

Prompt 3 1=SAVE MEANINGS THRU ALL PROGRAMS?
Respond EXEC if you are not sure.
or 1 and EXEC if variable meanings are described and are to be carried from program to program.

Prompt 4 For annotation KEY -- 0=Summary only 1=In Listing 2=In Margin DEFAULT=0 ?
Respond EXEC if you are not sure.
or 1 and EXEC to get meanings embedded in the listing.
or 2 and EXEC to get meanings placed in the right margin of the listing.

Prompt 5 PAPER WIDTH NARROW=0 WIDE=1 DEFAULT=0 ?
Respond EXEC or 0 and EXEC for 80 column listings
or 1 and EXEC for 120 column listings

Prompt 6 List mode KEY--0=List + XRef 1=List only 2=XRef only DEFAULT=0 ?
Respond EXEC or 0 and EXEC to get a complete listing.
or 1 and EXEC to get a listing of only the program text,
or 2 and EXEC to get a listing of only the cross-reference summary.

Prompt 7 START STMT DEFAULT=0 ?
Respond EXEC to get a listing of all lines of the program.
or 1 and dddd EXEC to get a listing beginning at a specified line.

Prompts 8 and 9 are repetitive through a list of file names.

Prompts 8 and 9 are repetitive through a list of file names.

```
Display   1 =310    2 =B10    3 =320    4 =B20
Prompt 8 Disk surface   Key single digit or any valid disk address 1__
Respond  EXEC
   or    digit and EXEC
   or    disk address and EXEC
Display   1 =310

Prompt 9 Disk Program Name + commentary. S.F.'1 ON ERROR 80)? __________
Respond  Program name (8 characters) plus commentary to append to title.
           and EXEC   prompt 9 will repeat until a single EXEC is entered.
  or     DATA and EXEC to get a list of program names from a list of DATA
         statements which may loaded to overlay the code at 9000+.
  or     ALL  and EXEC to localize the search by a common 0-8 byte IDentity.
  or     ALL- and EXEC to exclude modules with a common 1-8 byte IDentity.
  or     '0 to select an alternative disk surface.
  or     EXEC       to terminate the list of items.
```

NOTE: once the prompting sequence has gotten to Prompt 8 or Prompt 9 the S.F. keys are very useful to pick-up within the sequences.
S.F. ' 0 may be used to select a new disk surface,
and S.F. '1 to pick-up in the list of file names.

- - - - - - - - - - - - - -   - - - - - - -   - - - - - -

Output:

Section 1 - An index page to the program files listed, disk surface and description.

Program listing:
Titled and page numbered listings of program files.
Section of decompressed program code.

Summary cross-reference section
Variables

Special functions

Statement numbers.

- - - - - - - - - - - - - -   - - - - - - -   - - - - - -

The cross-reference program will create a listing of any BASIC program file. The listing will consist of titled and numbered pages. Each listing contains a program listing where each segment of a multi-statement line appears on a separate printer line. The listing will contain a blank line after each GOTO, RETURN, LOAD, or STOP statement. The listing will be automatically indented following IF statements, if not followed by another IF statement, and following FOR statements.

A separate summary section lists:

1) # references and variables used, meanings (optional), and referencing statement lines.
2). Special functions and meanings (optional).
3). Statement number and special function cross-references.

SPECIAL OPTION -- PAGE FORMATTING:

The CROSSRF program is written to interpret certain REM statements in a way that will enhance program documentation. REM% statements may be incorporated into the source BASIC causing speical listing effects.

REM% - REM (per cent) (up arrow) (comment)
positions to a new printer page with title and expanded print comment.

REM%d- REM (per cent) (digit 1-9) (comment)
skips n lines and prints the comment in expanded type.

REM% comment - REM (per cent) (comment)
skips 2 lines and prints the comment in expanded type.

:REM/ comment - (colon) REM (slash) (comment)
puts the comment in the right column of the previous statement.

SPECIAL OPTION -- ENTRY OF MEANINGS TO VARIABLES AND SPECIAL FUNCTIONS:

A descriptive meaning may be assigned to each variable, # file reference, or special function. This meaning, if used will be automatically output in the cross-reference summary listing. The meaning may also be output in the body of the listing either embedded into the program code or in the right margin on each occurence of the variable or special function. The descriptive material is entered into special tables by special REM statements. The special statements, described in the below, may be entered either in the program text or in a separate program file. Any variable may be given a new meaning merely by entering a new meaning item. Refer to question 3 in the interactive sequence for activating the meanings option.

REM%0 -REM (per cent) (zero) causes an entry to be placed into the meanings table. Meanings entries are coded as follows:

name space description comma
or name space description colon
or name space description carriage return
where name is up to 4 characters with no spaces, eg A0$(
space is one or more spaces,
description is 1 to 16 characters of description (no commas)
comma, colon, or carriage return terminates the meaning.

Example of BASIC program code required to enter meanings.

```
30 REM%0 A A, B CRT line, C Start char pos, C$ C$1, L Field size
40 REM%0 '201 Edit input,  #5 CRT
```

TABLE CAPACITIES

The CROSSRF program does not use any storage other than CPU memory. In order to create cross-reference summaries it is necessary to store certain table in system memory. Memory space for these tables is coded into COM statements in the loader module (TBO.CRF0). A separate set of COM statements for each memory size has been incorporated on the program disk. The default values for COM statements of various memory configurations may be modified if necessary.

Function: Program or system comparison
Modules: TBO.CMPS
Equipment used: CRT / keyboard, disk and optional printer.

Abstract: Program files on any system disks are compared line by line. Differences between the two files are displayed line by line. Two system disks may be compared by activating a list of DATA statement names.

The program comparison utility operates in one of two modes:
1). program mode allows the comparison of single program files. The programs may have the same name on different disk surfaces or different names on the same disk surface.
2). system mode allows the comparison of a list of program files occuring on two different disk surfaces. The list of names is from DATA statements overlaid over lines 9000 and beyond. The operator keys DATA when prompted for the first input file name.

Output to the CRT is scrolled and will show:

```
#1 IS ON LINE = 0010
                                        #2 IS ON LINE = 0010

#1 IS ON LINE = 0020
                                        #2 IS ON LINE = 0020

#1 --
      0020  COM @Z$(26)80, @X$(2)2, ... text for line with differences.

#2 --
      0020  COM @Z$(26)80, @X$(1)2, ... text for line with differences.
```

If output is to the CRT only, the program will pause at each differing line. The operator may key any key to continue the search. Keying RUN will allow the program to continue the search of the current files without pause.

If output is to the printer the program will continue without pause through the entire list of programs to be compared.

Operating instructions:

Display	PROGRAM COMPARISON
	NOTE-- TO COMPARE SYSTEMS WITH INPUT VIA DATA STATEMENTS
	ENTER THE NAME DATA AS THE 1ST PROGRAM NAME FOR #1
	Key S.F. '1 to pickup in entry of program names
prompt	KEY 0 if output to CRT?
respond	0 and RETURN for output to the CRT
or	1 RETURN for output to Printer

additional prompts for printer output only

prompt	# OF COPIES?
prompt	COMMENT ?
respond	comments for up to 9 lines of commentary for the top of the listing.
or	space and RETURN to end the COMMENTS.

display	FIRST PROGRAM---#1
	1 =310 2 =B10 3 =320 4 =B20
Prompt	Disk surface Key single digit or any valid disk address 1__
Respond	EXEC
or	digit and EXEC
or	disk address and EXEC
Display	1 =310
prompt	PROGRAM #1 NAME? ________
respond	8 character program name and EXEC
or	DATA and EXEC

display	SECOND PROGRAM--#2
	1 =310 2 =B10 3 =320 4 =B20
Prompt	Disk surface Key single digit or any valid disk address 1__
Respond	EXEC
or	digit and EXEC
or	disk address and EXEC
Display	1 =310

	the prompt below will occur if DATA was not entered for PROGRAM #1.
prompt	PROGRAM #2 NAME? ________
respond	8 character program name and EXEC
or	DATA and EXEC

Function: Analyze data file structure

Modules: TBO.ANDF

Abstract: This program analyzes a catalogued data file for structure. The output is a summary of the SAVE statements used to create the file.

```
Example  File name=@SYSFILE
         File size = 32 Sectors.
         File=@SYSFILE   BASIC-2 data structure -- SAVE number 1
         $8    Scalar   $32   $( 2)16  $(16)8  $(33)3   $(15)13

Example  File name=BSC*010A
         File size = 40 Sectors.
         File=BSC*010A   BASIC-2 data structure -- SAVE number 1
         $( 256)16
         File=BSC*010A   BASIC-2 data structure -- SAVE number 2
         $( 256)16
         END OF FILE
```

Operating instructions:

```
Display      Analyze Data File Format

         $(4)62                       Wang T.C. file format
         $2, $3, $(4)60               Wang Prom file format
         $(256)16                     4K TC microcode
         $(512)16                     8K TC microcode
         $(34)121                     4K TC ucode (min.disk storage)
         $(3)83                       3270 Spooler names file
         $(3)80                       3270*PQ
         $31, $(128)2, Scalar(3)      VFU format tape
         $8, Scalar, $32, $2(16), $(16)8, $(33)2   @SYSFILE

         Key '15 to see file examples
          1 =310  2 =B10   3 =320  4 =B20
Prompt   Disk surface  Key single digit or any valid disk address 1
Respond  EXEC
   or    digit and EXEC
   or    disk address and EXEC
Display   1 =310

display  see example above.
```

Function: File Z A P

Modules: TBO.ZAP

Abstract: A single disk surface may be examined. The display shows the contents of a single sector in hexidecimal notation and in ASCII.

1. Items in the catalog index area are flagged as AP (Active Programs), AData (Active Data).
2. Data files are displayed with highlighted attribute bytes.
3. Program files are displayed with highlighted RETURN codes.

Caution, you have the ability to change any all data in a selected sector.

The CRT will show a full screen non-scrolling display of the form below. The contents of the boxes pictured as (alpha) are described in the writeup below.

```
-----------------------------------------------------------------------------
!                !        !                     !              !            !
!      (A)       ! (B)    !         (C)         !     (D)      !    (E)     !
-----------------------------------------------------------------------------
!                        !  Wang Laboratories, Inc.                          !
!      (F)               !   SUPERZAP  Ver. d.d.d.                           !
!                        !                                                   !
-----------------------------------------------------------------------------
!         !                                 !             !           !
!         !                                 !             !           !
! (G)     !           (H)                   !     (I)     ! (J)       !
!         !                                 !             !           !
!         !                                 !             !           !
!         !                                 !             !           !
!         !                                 !             !           !
!         !                                 !             !           !
!         !                                 !             !           !
!         !                                 !             !           !
!         !                                 !             !           !
!         !                                 !             !           !
!         !                                 !             !           !
-----------------------------------------------------------------------------
!                              (K)                                          !
!                                                                           !
-----------------------------------------------------------------------------
```

(A)	Sector = ddddd	contains the disk sector number being examined.
(B)	Device = hhh	contains the disk surface being examined.
(C)	(alpha text)	Shows a MODE: eg. blank, Loading, printing, display
(D)	(alpha text)	Prompt entry field.
(E)	(alpha text)	Prompt entry field.
(F)	(alpha text)	Disk sector description: catalog, Program, Data file.
(G)	(alpha text)	In the catalog shows AP and AData items. In a named catalogued data file shows variable sizes. In a named program file shows the initial line number.
(H)	(hex values)	Values in hex of the sector being examined. In a named catalogued data file highlighted SOV bytes. In a named program file highlighted Carr.Return bytes.
(I)	(ASCII text)	Values in ASCII of the sector being examined.
(J)		
(K)	Description of possible operations.	

'0 Change sector
'1 Change device

'3 HEX to decimal
'4 Exit program

'9 Find ASCII string
'10 Find HEX string
'11 Find start of file
'12 Next sector
'13 Previous sector

'14 Print sector
EDIT=Enter edit mode

(K) Prompts for File Z A P

Description of possible operations.

prompt '0 Change sector
respond dddd and RETURN valid disk sector address.
or

prompt '1 Change device
respond hhh and RETURN valid 3 hexdigit disk address.

prompt '3 HEX to decimal
respond hh and RETURN valid hex values.

'4 Exit program

'9 Find ASCII string
respond text and RETURN valid ASCII data to find.

'10 Find HEX string
respond text and RETURN valid HEX data to find.

'11 Find start of file
respond valid file name and RETURN.

'12 Next sector
'13 Previous sector
'14 Print sector

Map Disk for Program Call Integrity

Abstract: From the "START" module this function analyzes a disk surface to find and correlate all program and data file references. Program modules found are of two types: menu modules are a list of DATA stattements origined at BASIC-2 line numbers 9000 and above. Function modules are all other program modules are called either from menus or other programs. The program operates in two parts. Part one "TBO.DMAP" goes through the menu structure and creates two COMmon tables listing menu modules and non-menu program modules. Part two "TBO.FMAP" goes through all the program function modules to find all program LOAD and LOAD at DATA LOAD OPEN statements.

Equipment used: CRT / keyboard and optional printer.

Operating instructions:

```
Display                      Function Analyzer
        Source Program disk -
         1 =310    2 =B10    3 =320    4 =B20
Prompt 1 Disk surface   Key single digit or any valid disk address 1
Respond  EXEC
   or    digit and EXEC
   or    disk address and EXEC
Display   1 =310
         List of programs is from COMmon P$() array
          1 START
          2 @CLOCK

          ...

         PROGRAM NAME?
Respond  EXEC
   or    program name and EXEC
         KEY 1(EXEC) FOR HARD COPY?

Display   (High-lighted Program name)    ..SELECTED LINES Prog. 1 of 166

         (listing of all LOAD statements found in program)

          (High-lighted Program name)    VERB=LOAD   IS USED d TIMES
         Files referenced
         P Menu     0001 STARTTBO
         PROGRAM LOAD is at 0010
         LOAD is at --0010

Example  - . - . - . - . - . - . - . - . - . - . - . - . - . -
         STRT3275           ..SELECTED LINES Prog. 28 of 166

              0040 DATA LOAD DC OPEN T #0, "M-3270"

              0060 DATA LOAD DC #0, W$()

              0150 LOAD DC T "EM3275"

           STRT3275           VERB=LOAD   IS USED 3 TIMES
         Files referenced
         D                  M-3270
         P Function 0177  EM-3275
         PROGRAM LOAD is at 0150
         DATA LOAD OPENs at 0040
         LOAD is at --0040--0060--0150

                    - - - - - - - - - - - - - - - - - - - - -
```