A REPORT from the

Cooperative Institute for Meteorological Satellite Studies

DOCUMENTATION OF THE VAS DATA PROCESSING SOFTWARE
DOCUMENTATION OF THE
VAS DATA PROCESSING SOFTWARE

by

James P. Nelson, III

Cooperative Institute for Meteorological Satellite Studies
Space Science and Engineering Center
University of Wisconsin
1225 West Dayton Street
Madison, Wisconsin 53706

November 1984
Preface

This manual summarizes the main and auxiliary software used to process VAS (VISSR Atmospheric Sounder) retrievals at the Space Science and Engineering Center at the University of Wisconsin-Madison. Since the software is being developed in a research laboratory, it is constantly undergoing change to upgrade and improve it. The programs described herein, with the exception of VTPZ, are versions from May 1, 1984. The version of VTPZ contained within the text is valid for May 3, 1984.

If there are any questions or suggestions, please mail them to:

Mr. James P. Nelson, III  
Space Science and Engineering Center  
University of Wisconsin  
1225 West Dayton Street  
Madison, Wisconsin 53706
# TABLE OF CONTENTS

I.) Introduction ................................................................................. 1

II.) Chapter 1: General Retrieval Outline ................................. 2

III.) Chapter 2: VAS Modular Flowcharts ............................... 18

IV.) Chapter 3: Main VAS Retrieval Software ......................... 47

A.) VPVA .............................................................. 49

Direct McIDAS to given sounder area.

B.) IDVA .............................................................. 52

Initialize VASTEXT documentation file and retrieval MD file row header.

C.) LOVA .............................................................. 63

Look at current version of VASTEXT file for user's terminal.

D.) SPVA .............................................................. 68

Set different MD file/row numbers, grid file/row numbers, etc. in VASTEXT file.

E.) GSVA .............................................................. 72

Prepare upper air first guess MD file, using data from first guess gridfile.

F.) CSVA .............................................................. 82

Calculate surface dewpoint depression, sea level temperature and 1000 mb height for surface stations and place in surface MD file.

G.) SRVA .............................................................. 91

Generate grids of the parameters calculated by CSVA.
H.) XRVA ........................ 102
Delete values (or entire reports) within
surface or retrieval MD files (is the editing
program for VAS retrieval processing.
I.) VTPZ ............................. 116
Carry out VAS retrievals via physical matrix
inversion algorithm (is the operational VAS
retrieval program).
J.) PLVA ............................. 142
Plot VAS retrieval or surface MD file data
such as heights, temperatures, etc. on
color-video display monitor.
K.) BNVA ............................. 156
Generate grids of heights, temperatures,
etc. from data taken from retrieval MD or
other types of MD files.
L.) GWVA ............................. 179
Generate wind vectors (geostrophic,
gradient, isallobaric or ageostrophic) from
height grids produced by BNVA.
M.) UGVA ............................. 189
Update VAS first guess gridfile using
previously-generated VAS retrievals.
V.) Chapter 4: Supplementary VAS Retrieval Software 195
A.) EXVA ............................. 196
Examine retrieval profiles in terms of
retrieved temperature and differences in
temperature between retrieval and first guess.

B.) VDVA .......................... 203

Display brightness temperature, navigation,
spin budget, radiance or filter number
information for the field of view at the
cursor or keyed-in location.

VI.) Chapter 5: Miscellaneous VAS Retrieval Software  207

A.) SRAD .............................. 208

Prepare retrieval box-averaged brightness
temperatures and other quantities for subsequent
use in SRET, the VAS iterative retrieval
program.

B.) SRET .............................. 228

Carry out VAS retrievals via an iterative
retrieval scheme (the former VAS operational
retrieval program).

C.) GPVA .............................. 265

Display VAS guess profiles of temperature,
dewpoint and mixing ratio at either the
cursor location or a keyed-in latitude/longitude
position.

D.) ESVA .............................. 271

Edit surface reports at cursor location,
either by deleting or substituting in any
combination of 1000 mb height, mean sea level
temperature or surface dewpoint depression
at a previously-existing report location,
E.) VTPW ........................................ 282

Retrieve total precipitable water vapor values for individual fields of view within a retrieval area, and (optional) generate an image of the results.

VII.) Acknowledgements .......................... 302

VIII.) Appendix I: General McIDAS Terminology .......... 303

IX.) Appendix II: Level I Flowchart Constructs ............ 329

X.) Appendix III: Subroutine List and Descriptions ....... 331

XI.) Appendix IV: Function List and Descriptions ........... 352

XII.) Appendix V: Miscellaneous Program Information .. 358

XIII.) Appendix VI: VAS Retrieval Software ............... 362

XIV.) References .................................. 487
Introduction

A large volume of software has been developed and written at the University of Wisconsin-Madison for the purpose of preparing and processing VAS radiances data and then retrieving profiles of atmospheric temperature and moisture. A person new to the Man-computer Interactive Data Access System (McIDAS) and/or VAS retrieval processing would have a difficult time assimilating and becoming familiar with the software without some type of guidance. This document is intended to alleviate this problem by explaining how the VAS software is operated in terms of McIDAS commands, program flowcharts and program descriptions. The manual does not explain all of the VAS software; rather, it covers only those programs of major importance to the retrieval process.

The text consists of five chapters and six appendices. The first chapter gives a broad overview of the retrieval process. The second chapter presents each program in terms of modular flowcharts. The final three chapters deal with individual programs within the retrieval process itself. These chapters involve three levels of importance, ranging from the most vital software to programs which are rarely used or have been deactivated. Chapter 3 details the software used on an every day basis, while Chapters 4 and 5 contain descriptions of several lesser used retrieval support and other discontinued programs.

The final section of the reference contains six appendices. Appendix I contains descriptions of different McIDAS terms used in the manual, such as sounder area, MD file, etc. Appendix II describes constructs which exist in one type of flow chart contained within the manual. Appendices III and IV contain lists of all program subroutines and functions, including a description of what each subroutine and function does. The fifth appendix describes program code length, number of accesses to storage (I-O), etc., for each program. Finally, Appendix VI is comprised of VAS retrieval software reproductions.
CHAPTER 1

General Retrieval Outline

In this chapter, a general summary of the VAS retrieval process is presented. Included are both a description of the steps undertaken when processing an area of retrievals in real time and a flowchart outlining these steps, as well as a corresponding series of sample processing commands. The flowchart follows directly on the heels of the retrieval process description, and provides a compact summary of the retrieval scheme.

Optional programs within the flowchart are denoted as such. Otherwise, if there is no attendant "optional" note, the user should assume that a given program is required for the retrieval process. In addition, the text description will differentiate between general McIDAS (system) programs and VAS retrieval processing programs by labelling the system programs as such.

Furthermore, text that relates to a given sample processing command will be denoted by the number of the corresponding command (in parenthesis). Since there are a total of 30 commands, there will be 30 corresponding numbers located at the proper locations within the text.

It should be noted that although this discussion deals with the processing of an area of retrievals, only minor modifications in the keyin of retrieval program VTPZ are required to process only a single retrieval. In addition, this discussion assumes the user has a McIDAS terminal complete with a television (video) monitor. Again, only minor changes in program commands are needed
to generate retrievals without a video monitor (known as the "auto" processing mode). In order to process retrievals in this mode, however, the user must know where VAS data exist in terms of the sounder area's NW and SE corner latitude/longitude coordinates. Any differences between the general discussion given here and the AUTO mode will be duly noted in the appropriate locations within the sample processing commands. Finally, to get help with any program when examining different options to the ones given here, simply key in "HELP program name," or examine the program within the McIDAS User's Manual.

Initially, the radiance data for the retrieval(s) the user wishes to process must be listed using system program LA and then (optional, but highly advisable) loaded into one or more of the user's sounder area(s) using system program AA (1). Following this, system program DF is used to display the band 8 (window) digital radiance data for the given area in a specified image frame on the television (video) monitor (2). In addition, it is also possible to plot a satellite-projection map over the image using system program IC (3). After the band 8 image has been displayed, the sounder area corresponding to the radiance data just displayed must be pointed at using VPVA (4). This has the effect of telling the system what radiance data it is to use for the retrieval processing. Then, as a quick check of the data quality, one can use VDVA, which can be keyed in to list brightness temperatures at the cursor location for all twelve bands used in the retrieval process (5). (Note: band 11 is usually not used, due to excessive noise, so its brightness
temperature will be displayed as 999999).

Up to this point, we have dealt only with the raw VAS radiance data. Retrieval processing of VAS radiance information also requires both surface data (if the surface data option is being used) and upper air guess data. This data is stored in both Meteorology Data (MD) files and gridfiles. In addition, the user must have a retrieval MD file to store the results of the retrieval processing, as well as a "scratch" gridfile for temporary storage of grids of retrieval parameters produced by program BNVA. One should check to see if these files exist using system programs MDU and IGU (6). If the files don't exist, they can be created using the same system programs (7). After the files have been made, the upper air guess data is loaded into the upper air guess gridfile using system program NMCU (8). At this point, the following files should exist: sounder area (loaded by AA), surface data MD file (schema: RSVC; for 1000 mb heights, sea level temperatures and surface dewpoint depressions; no data yet), upper air guess MD file (schema: VGSS; no data yet), retrieval MD file (schema: VRET; no data yet), upper air guess gridfile (loaded by NMCU), surface gridfile (no data yet), and the scratch gridfile (no data yet).

The next required step is IDVA, whose function is to initialize the VASTEXT file and the retrieval MD file row header (9). The VASTEXT file can be envisioned as a bookkeeping file whose purpose is to keep track of many different retrieval parameters, such as retrieval MD file number, sounder area, retrieval type, etc. The VASTEXT file contents can be displayed
on the CRT at any time by using program LOVA (10).

The next steps involve IGU again and system program IGG. IGU is used to set the grid file pointer to the upper air guess gridfile (11), and IGG is used subsequently to list the grids contained within this gridfile (12). This step is necessary to pick out the grids containing the most suitable guess for the sounder area being processed. Usually, the most suitable guess will consist of the grids closest in time to the sounder area you wish to process. Finally, the gridded guess data is reformatted from the guess gridfile into the upper air guess MD file via GSV (13). In other words, values from numerous grids (up to 22, consisting of 15 temperature grids, 6 dewpoint grids, and one 1000 mb height grid) for a given gridpoint are stored at a certain row and column coordinate in the guess MD file. This completes the upper air guess data processing.

The next stage of the retrieval processing scheme involves the surface data. This data is prepared using programs CSVA and SRVA. CSVA calculates 1000 mb heights, sea level temperatures and surface dewpoint depressions for each station and places the results in the designated surface MD file (14). Then, SRVA takes this data, generates a grid of one of the parameters, and places the results in a grid location within a designated surface gridfile (15).

The results of the surface data grid analysis can be examined by first setting the gridfile pointer to the number specified in the SRVA command using system program IGU (16), and then using system program IGT to contour the desired parameter.
on the video screen (17). If bad analyses of any or all three surface parameters result, the user should first clear the graphics screen and then re-plot the satellite projection map over the image (using system programs EG and IC), and then go back and plot the surface data with PLVA (18). Then, unacceptable reports can be deleted with XRVA (19). Note the MDU SET before the execution of XRVA. Following this, the process from SRVA forward must be done again to generate corrected grids of any or all of the three surface parameters (20).

With the completion of the surface analyses, it is now possible to do the actual VAS retrievals. However, before the retrievals are made, set the "NO. RETRIEVALS=" value in the VASTEXT file to 0 by using program SPVA (21). This will ensure that the correct number of retrievals about to be performed is stored in the VASTEXT file. Then, the retrievals can be made by executing program VTPZ (22). This program generates profiles of temperature and moisture, and also creates an image of total precipitable water vapor. Note the example of the VASTEXT file as it stands after the area of retrievals has been processed.

Up to this point, it has been possible to carry out the retrieval processing via either a video or non-video terminal. However, since the emphasis from now on lies mainly with visual retrieval editing, only a video terminal should be used.

After the retrievals have been generated, various retrieval parameters (such as height, temperature, etc.) can be viewed on the video screen using program PLVA (23). Then, before grids of retrieval parameters are generated by using program BNVA, IGU
should be executed again to set the gridfile pointer to wherever the user desires the results of BNVA to be placed (24). Usually, the retrieval parameter grids are stored in a different gridfile (the scratch gridfile) than either the first guess or surface gridfiles. At this point, BNVA is executed, with the results being stored in the scratch gridfile (25).

Since one of the statements outputted by BNVA indicates into which grid of the gridfile the analysis was stored, IGTW can be used again in the same fashion as with the surface data to contour various retrieval data grids (26). Poor analyses can then be corrected in the same manner as the surface data analyses (EG, IC, PLVA, MDU and XRVA) (27). If some deletions are done, the user must repeat the steps from BNVA (step 25) onward to assess how the deletions affect the gridded contours (28).

Finally, VAS gradient winds can be produced and plotted using GWVA (together with height grid(s) generated previously by BNVA) (29). Note that the grid number(s) in which the gridded height results of BNVA were placed WILL HAVE CHANGED if any deletions were done (see step 28 above). This is so because BNVA will place updated retrieval grids (after deletions have been done) in positions after the original unacceptable grids. (In the sample processing commands associated with this discussion, it was assumed that no deleting was necessary and that therefore the 500 mb height grid was grid number 1.)

Lastly, if retrievals from a different time period must also be processed, a more up to date (closer in time) first guess can be generated using the just-completed retrievals via program UGVA.
(30), and the entire procedure begins anew.

On the pages following the flowchart, a sample of McIDAS keyins for the processing of a typical retrieval area is presented. These commands correspond directly to the above text.
VAS RETRIEVAL PROCESS FLOWCHART

"#" MEANS DOCUMENTED WITHIN THIS MANUAL

LA
AA
DF

VPV

MDU

IGU

NMCU

IDVA

igu

IGG

GSVA

CSVA

SRVA

IGU

IGTV
Summary of Files Used

Note: assume the user has been assigned sounder areas 4176-4191, MD files 4176-4191, and gridfiles 4176-4191.

A.) sounder area (data at 1248 GMT, 21 MAY 84): 4190
B.) digital area for storage of total precipitable water vapor image (created by retrieval program VTPZ): 4191
C.) surface data MD file (schema: RSVC; containing Z100, TSL and DD): 4183
D.) upper air guess MD file (schema: VGSS): 4184
E.) retrieval MD file (schema: VRET): 4185
F.) upper air guess gridfile: 4190
G.) surface grid file (containing grids of Z100, TSL and DD): 4189
H.) scratch gridfile for temporary storage of retrieval grids: 4191

Command Sequence

1.) Locate, and move radiance data:

   LA 1300 1310 (lists GOES-EAST realtime VAS sounder areas)

   then:

   AA 1305 4190 ASIZE=ALL (assume area 1305 contains VAS radiance data from 1248 GMT, 21 MAY 84)

2.) Load digital sounder area into given image frame on television monitor:

   (** in AUTO mode, numbers 2 and 3 are not done ***)

   DF 4190 1 EC 35 90 2

3.) Draw map over image (satellite projection):

   IC
4.) Point to correct sounder area:
   VPVA 4190

5.) Display brightness temperatures at cursor location:
   VDVA B

6.) Check to see if necessary MD and gridfiles exist:
   MDU LIST 4176 4191
   IGU LIST 4176 4191

7.) Create MD and grid files (if the files do not exist):
   MDU MAKE 4183 RSVC 1 X X 84142 "SFC DATA 12GMT 21 MAY 84
   MDU MAKE 4184 VGSS 1 X X 84142 "U/AIR GUESS 00GMT 21 MAY 84
   MDU MAKE 4185 VRET 1 X X 84142 "VAS RET. 1248 GMT 21 MAY 84
   IGU MAKE 4189 "EDITED SFC DATA 12GMT 21 MAY 84
   IGU MAKE 4190 "LFM FROM 84 05 21 00
   IGU MAKE 4191 "SCRATCH GRIDFILE FOR RET. GRIDS

Note: Text labels after quotation marks are optional (but recommended) and only for the user's benefit.

8.) Load current upper air guess into guess gridfile:
   NMCU 0 4190 FILE=LFM000Z "LFM GSS FROM 00GMT 21 MAY 84

9.) Initialize VASTEXT file and row header of retrieval MD file for displayed image:
   IDVA 4185 1

   (*** in AUTO mode, IDVA 4185 1 45108 35080 AUTO=1, as an example, having specified the NW and SE corners to be covered as 35 to 45 degrees N latitude and 80 to 108 degrees W longitude ***)

10.) Check VASTEXT file contents:
   LOVA

11.) Set gridfile number to upper air guess gridfile:
IGU SET 4190

12.) List grids in gridfile 4190:

IGG LIST Note: grid 33 usually starts sequence of 12-hour forecast grids.

13.) Reformat guess from guess gridfile 4190 into guess MD file 4184, for the 12-hour forecast sequence, starting at grid 33:

GSVA 4190 33 4184

14.) Calculate Z100, TSL and DD for each station in real time surface hourly observation MD file (one of MD files 1-10, schema = SVCA), and load into surface MD file (schema = RSVC) 4183:

GSVA 4183 Note: hour of surface data will be 12 GMT (truncation of satellite observation time) and, thus, row 12 in MD file 4183 will be used to store the data.

15.) Create one grid each of Z100, TSL and DD (data taken from surface MD file 4183) and store in grid positions 1-3 in surface gridfile 4189:

SRVA Z100 4189
SRVA TSL 4189
SRVA DD 4189

16.) Examine results; first set gridfile to 4189:

IGU SET 4189

17.) Contour data in gridfile 4189:

IGTV 1 20 SAT (Z100--grid number 1) (***(in AUTO mode, numbers 16-20 are not done ***)

IGTV 2 3 SAT (TSL--grid number 2)

IGTV 3 3 SAT (DD--grid number 3)

18.) If bad analyses result, examine surface MD file. First,
clear the graphics screen:

EG, followed by

IC (draw satellite projection map over image)

then:

PLVA Z100 MDNR=4183 MDRR=12 for Z100 (with MDNR, MDRR indicating MD file and row numbers for surface data, respectively)

then:

PLVA TSL MDNR=4183 MDRR=12 for TSL, and

PLVA DD MDNR=4183 MDRR=12 for DD

19.) Delete bad reports and update surface MD file:

MDU SET 4183

then: (**in AUTO mode, numbers 16-20 are not done ***)

XRVA (delete ENTIRE report)

20.) If any deletions, repeat (for that parameter) starting at step 15.

21.) Set "NO. RETRIEVALS=" in VASTEXT file to 0:

SPVA NRET=0

22.) Process retrievals for the sounder area:

VTPZ END=3513 6997 SPC=16 11 ARA=4191

(**in AUTO mode: VTPZ SPC=16 11 ARA=4191 AUTO=1, as an example ***)

In the non-auto mode, the desired ending location (in image coordinates) is determined by setting the cursor near the SE corner of the image and then executing system program I (simply key in the letter "I" by itself).

The starting location will be determined by default
from the VASTEXT file. The retrievals generated will be for 11 x 11 FOV (field of view) boxes (default), separated by 16 FOV's in the y-direction, and 11 FOV's in the x-direction. In addition, a total precipitable water vapor image will be created and stored in digital area 4191. The following example shows what the VASTEXT file should look like at this point. Remember, the VASTEXT file is displayed on the CRT by using program L0VA.

YYDDD BEGIN X-RES Y-RES LLNW LLSE STAT NSAT SNDAREA
84142 124800 8 8 47110 33078 1 29 4190
MDNS MDRS MDNG MDRG MDNR MDRR
4183 12 4184 0 4185 1
NGFG NGFS ZGRID TGRID DGRID PGRID
4190 4189 1 2 3 84142

NO.RETRIEVALS=183
CURRENT RETRIEVAL OPTIONS..

TYP GSS SPC SIZ SFC ENDL ENDE BEGL BEGE TER PLT
0 0 20 10 0 3421 7149 2417 4673 68 0

Note: The above information pertains to the processing of a sounder area sensed in the small detector mode of VAS. For the large detector mode, variables X-RES and Y-RES would be equal to 16.

NOTE: Beyond this point, only a video terminal should be used. This is so because the editing which follows is most efficiently performed on this type of terminal, where XRVA
is used with cursor selection to delete unacceptable retrievals.

23.) Plot retrieval data:
   PLVA Z 500 MDNR=4185 MDRR=1

24.) Set gridfile before BNVA:
   IGU SET 4191

25.) Generate grids of retrieval parameters and store in gridfile 4191:
   BNVA Z 500

26.) Select the appropriate grid (#1, because grid just generated was dumped here by BNVA), and display the contours on the video screen (over the earlier PLVA plot of 500 mb heights):
   IGTV 1 30 SAT (contour interval of 30 meters)

27.) If there are poor analyses, go through a similar procedure as was used for the surface data:
   EG
   IC
   PLVA Z 500 MDNR=4185 MDRR=1
   MDU SET 4185
   XRVA (delete entire poor retrieval)

28.) If there are any deletions from the retrieval MD file, repeat from step 25.

29.) Generate VAS gradient winds (optional):
   GWVA 1 500

30.) Update upper air first guess (i.e., grids 33-54 in gridfile 4190) with the just-completed retrievals, if more retrievals
are to be made at a different time, and then begin again at step 1 to process the next retrieval area:

UGVA 33 Note: the updated grids will be added onto the end of gridfile 4190.
CHAPTER 2

VAS MODULAR FLOWCHARTS

This chapter contains modular flowcharts for each VAS program discussed in chapters 3-5. The flowcharts are modular in that they consist solely of subroutines which are called successively in a given program. Each subroutine is therefore a module of the program as a whole. The programs are subdivided to systems level or assembly language subroutines in most cases, which is sufficiently detailed for the purposes of this manual.

The first tier of subroutines in each flowchart, which will be referred to as Level I Subroutines, are those routines called explicitly by the given program. Subsequent lower-level subroutines branch out from these main subroutines. When a given subroutine terminates a branch of the program "tree," it means simply that I could no longer trace that particular branch, or that it was not important to go any further.

Before a list of the order of flowchart presentation, I shall discuss briefly the page number system adopted for this chapter. When a given flowchart refers to a certain page number, the user should turn to the page whose SECOND (right) number at the bottom corresponds to the correct page. The user should note that there are 27 such double-numbered pages in Chapter 2.

ORDER OF PRESENTATION

1. VPVA
2. VDVA
3. IDVA
4. GSV
CHAPTER 3
Main VAS Retrieval Software

This chapter contains those programs most vital to the VAS retrieval process. These programs, in addition to the other supporting McIDAS software mentioned in chapter 1, are necessary to process VAS data sets completely. Much of the software discussed in this chapter and the two which follow can apply to either VAS or TOVS. However, I will deal exclusively with the VAS aspects of the programs.

The presentation of each program in this chapter and the two which follow is similar. First, a general description of the program and how it operates is given. This should help someone unfamiliar with the software to get at least a basic understanding of its function. Next, a different type of flowchart than the Modular Flowchart, called a Level I Flowchart, is presented. In addition to Level I Subroutines, these flowcharts include comments, DO LOOPs and any IF statements that affect the calling of a Level I Subroutine. Finally, an attempt is made to define each of the variables within the IF statements themselves with at least one arithmetic statement, and more than one if the value of the variable changes within the program proper. The constructs used in the flowcharts are described in Appendix II. To gain the fullest understanding of each program, the Level I Flowchart should be used concurrently with both the program description and its code reproduction in Appendix VI.

The final section of each program discussion consists of a list of the subroutines called within that program. Level I
Subroutines are indicated at the head of each list with a capital "I." This list should correspond exactly to the subroutines included in the program's Modular Flowchart in Chapter 2.
VPVA

This program directs McIDAS to the sounder area/sounder file containing VAS radiances for the retrieval or retrievals one wishes to process. The area number is stored in the variable IFILE. In addition, if 0 .LT. IFILE .LT. 9999, User Common (UC) word 81 is set to the area value, and a directory for that particular sounder file is read. The directory contains information about the file, such as satellite number, date, etc. A call to subroutine OUTINT (CALL OUTINT (KOUT)) will then display this information on the CRT.

On the other hand, if either a blank or 0 is keyed in for the sounder area, the sounder area pointed to will be the current UC value in word 81 (200 NFILE=LUC(81)). In addition, the directory information (assuming NFILE .GT. 0) will again be displayed.

Finally, if the keyin is "VPVA -1," the pointer will be set to sounder area 0, and the system will then be set for TOVS processing. No directory information will be displayed using this option.
SET USER COMMON
WORD 31 TO KEYED-IN
SOUNDER AREA NUMBER
(assuming, of course, that
SOUNDER AREA NUMBER
has been keyed in)
Subroutines used by VPVA:

1) IQ-I
2) PUC-I
3) TQME-I
4) READD-I
5) OUTINT-I
6) PUTCJR
7) WD
8) BLKA
9) STC
10) ITOC
11) MOVBl
12) LTD
13) CLEANA
14) TQ
15) PRDLNX
16) PRLOS
17) PROPEN
18) PRPRPR
19) LOCK
20) PRWR
21) PRRD
22) UNLOCK
23) PRCN
24) POST
25) ABORT
26) II
27) EDEST
28) MOVW
29) MOVCW
30) CLEANW
IDVA

IDVA is used to initialize the VASTEXT documentation file and a VRET-schema retrieval MD file row header. The discussion which follows will deal with two modes of operation. In the first mode, I will assume no latitude/longitude boundaries for processing have been keyed in. In the second mode, known as the "auto" mode, I will assume keyword parameter AUTO has been keyed in to some non-zero value, and I will also assume that both the NW and SE image corners have been keyed in via positional parameters LALONW and LALOSE. The first mode is used when image information is available to the user; the second mode is used when no image information is available, but the NW and SE corners of the area to be processed are known.

In the first mode case, variables MDNR, MDRR and KBUG (retrieval MD file number, retrieval MD file row number, and debug option, respectively) are read in via positional and keyword parameters (MDNR=IPP(1,0), MDRR=IPP(2,0), KBUG=IKWP('BUG',1,0)). Then, after the VASTEXT file is opened and read, control passes to statement 20 and then immediately to statement 100 (note that LALONW=0 for this mode, since it was not keyed in). At this point, image information is accessed by subroutine GETFRM, including such things as the upper left corner line and element of the TV frame (IL,IE). Subsequent steps result in the picking up of the sounder area (NSND) and the calculation of the lower right corner of the TV frame itself in terms of line and element coordinates (LL,LE). Then, subroutine VASDAT is used to locate the upper bounds of the VAS data by moving successively
down the TV screen. At least 4 consecutive lines of data must be found before control will transfer to statement 130. This constraint guarantees that the maximum number of small-detector fields of view will be used when generating retrievals later on in VTPZ. After the upper bounds of the data is located (which gives the line coordinate of the NW corner of the area (image) to be processed), the leftmost boundary of the data (element coordinate of the NW corner) is located (see code between statements 133 and 135). At this point, control passes to statement 138, where a message saying "...UPPER LEFT DONE" is displayed on the CRT. Now, the same process is done for the lower right (SE) corner of the satellite image (find lower bounds of data, (giving the lower right line coordinate of the image), then locate lower right element coordinate of the image). This is done approximately between statements 140 and 157. Keep in mind that LALONW=0 during this entire process. After the SE corner of the WAS data has been successfully located, variables IE, LE, LOW and LOE are determined (IE=MIN(IETOP,IEBOT), etc.), and a message saying "...LOWER RIGHT DONE" is displayed on the CRT. Then, LALONW and LALOSE are calculated and stored in IDOC(25) and (26), respectively. Further steps initialize the retrieval MD file row header (assuming that MDNR and MDRR have been keyed in .NE. 0) and the VASTEXT file. The row header is initialized by putting the array IRET into the header via function MDPUT (I0K=MDPUT(MDNS,MDRR,0, IRET), while the VASTEXT file is updated by placing several pertinent variables such as the retrieval MD file and row numbers, status flag, sounder area number, spacing
defaults, etc. into the array IDOC, which is then written, via subroutine DWRITE, into the VASTEXT file itself. This concludes the first mode of operation of IDVA.

For the second "auto" mode case, IDVA proceeds identically to the first mode up to statement 20. Then, since both LALONW and LALOSE have been keyed in, IDOC(25) and (26) are filled, variables LAN and LOW are determined, and information pertaining to the sounder area is accessed via subroutine READD (CALL READD(NSND, IDIR)). Following this, after the navigation has been initialized (NVINIT), four calls to subroutine SATEAR, in conjunction with other NVINIT calculations, determine the line and element coordinates of the area to be processed (initial line (IL), initial element (IE), last line (LL) and last element (LE)). Note that no mention has been made of the TV image within the discussion of this mode, and that the boundaries of the area to be processed with this mode may not coincide with the actual sounder area boundaries, as they did for the non-"auto" mode.

After control transfers to statement 115, IDVA functions like the previously-discussed non-"auto" mode up to statement 135; that is, the upper bounds of the VAS data is located, giving the NW line coordinate, and then the western boundary is determined, giving the NW element coordinate.

At this point, however, the program functions differently than the first case, in that the element coordinate of the NE corner of the VAS data (LETOP) is calculated, something which is not done in the initial mode of IDVA. Then, after the "...UPPER LEFT DONE" message is printed on the CRT, the lower bounds of the
VAS data is calculated, followed by the determination of the SW and SE element coordinates. Now, IDVA is at statement 157 again, and it proceeds identically to the first mode from this point to the end.
READ IN RETRIEVAL NO.
FILE AND ROW, AS WELL
AS NW AND SE LAT/LON
COORDINATES

OPEN AND READ VASTEXT
FILE

THIS PATH WORKS ONLY
ON A VIDEO TERMINAL

BEGIN SEARCH FOR MAX AND
MIN LAT AND LONG REQUIRED BY
GREAT, TUNT FOR MESSING LINES
AT BEGINNING AND END OF GREAT

READ SOUNDER DIRECTORY
FOR SOUNDER AREA "NSND," AND INITIALIZE NAVIGATION

CALL LVSSL (GETAIN, BEDIT, INAV, PTIME)
LOOK AT ALL CORNERS
TO GET MAX AND MIN
ELEMS.

DATA NCOLS/56/, WLON/360/, ELON/90/, TPE/360/, BOTW/8/
DATA LETOP/8/, IEBOT/99999/
MONR = IPP(1, φ)
MORR = IPP(2, φ)
LALONW = IPP(3, φ)
LALOSE = IPP(4, φ)
KEBUC = KWP (BUG: 1, φ)
CALL TSNIO (1, 1, 1, 1, 100, 1)
LALONW = φ
CALL GOEST
( LALOSE HAS NOT BEEN SPECIFIED, φ)
END
CALLED LGT
( LALOSE ≠ φ)
28 φ
LALOSE = φ
CALL LGT
( LALOSE = φ)
FOR TOVS
FILL DIRECTORY AND EXIT
( NSND, IOIR)
CALL REDD (NSND, IOIR)
NODE = 1
USE NAV TO GET LINE, ELE
CALL TSNIO (1, 2, K, )
(LALONW, LON)
Subroutines used by IDVA:

1) DOPEN-I
2) DREAD-I
3) TSNIO-I
4) GETFRM-I
5) VASDAT-I
6) SDEST-I
7) OUTINT-I
8) DWRITE-I
9) DCLOSE-I
10) ENKODE-I
11) READD-I
12) NVINIT-I
13) SATEAR-I
14) ABORT-I
15) ENCODE
16) ENCODX
17) CONTNT
18) LTQ
19) MOV
20) CLEANA
21) TQ
22) PRLNX
23) PRCL
24) PROIX
25) PRPR
26) LOCK
27) PRWR
28) PRD
29) UNLOCK
30) PRCL
31) POST
32) BLKA
33) ZECONV
34) II
35) STC
36) MOV
37) TMES
38) ITOC
39) MOV
40) GETNAV
41) EPOCH
42) GETGAM
43) VASNAV
44) DDEST
45) CLEANW
46) EDEST
47) PLNKL
48) OPNA
49) LOPEN
50) LGEL
51) SATPOS
52) ANGGET
53) SOLARP
54) ANGLES
55) READOF
56) RBYTSX
57) ZEROS
58) RDTRK
59) MOV
60) MOVW
61) HROTOPO
LOVA

The purpose of LOVA is to display the contents of the VASTEXT file for the user's local terminal. As always for VAS applications, the sounder area must be set previously by VPVA.

First, the VASTEXT file is opened and read via subroutines DOPEN and DREAD, causing such retrieval processing parameters as sounder area number (NSND), retrieval MD file number, number of retrievals processed (NRET), etc. to be passed to LOVA through the array IDOC (call DREAD(LUN, ITERM, IDOC)). A further call to subroutine SDEST outputs data variable TITLE to the CRT, whereupon subroutine ENKODE outputs values corresponding to the list of variables in TITLE. Further calls to SDEST and ENKODE output still more of the VASTEXT file contents.

The list at the end of this brief discussion deals with all of the variables listed by LOVA. The program(s) in the parentheses to the immediate right of each variable indicate(s) which program(s) affect that particular variable. Furthermore, the element of the array IDOC which each variable occupies is also listed. Note that as one goes continually further in the retrieval process, executing more of the pertinent programs, more of these variables will be assigned values. Note also that program SPVA can be used at any time to change different VASTEXT file variables.

1.) YYDDD (IDVA-derived from IDOC(1), CSVA)- Julian date of VAS data
2.) BEGIN (IDVA-IDOC(2))- beginning time of VAS radiance measurements
3.) Y-RES (IDVA-IDOC(9)) - "Y" coordinate resolution of infrared measurements (in kilometers)
4.) X-RES (IDVA-IDOC(10)) - "X" coordinate resolution of infrared measurements (in kilometers)
5.) LLNW (IDVA-IDOC(25)) - latitude and longitude of NW corner of retrieval area to be processed
6.) LLSE (IDVA-IDOC(26)) - latitude and longitude of SE corner of retrieval area to be processed
7.) STAT (IDVA-IDOC(28)) - status flag (is set to 1 in IDVA, and must be such for CSVA to work)
8.) NSAT (IDVA-IDOC(5)) - satellite number (29 for GOES-EAST, 31 for GOES-WEST)
9.) SNDAREA (VPVA-IDVA-IDOC(35)) - sounder area currently pointed to
10.) MDNS (CSVA-IDOC(36)) - surface data MD file (containing Z100, TSL and DD data)
11.) MDRS (CSVA-IDOC(37)) - row of surface data MD file that pertinent surface data is contained in (time dependent)
12.) MDNG (GSVA-IDOC(38)) - guess MD file (upper air)
13.) MDRG (GSVA-IDOC(39)) - row number of guess MD file (as many rows as needed are filled, so this variable is usually not important)
14.) MDNR (IDVA-IDOC(40)) - retrieval MD file number
15.) MDRR (IDVA-IDOC(41)) - retrieval MD file row number
16.) NGFG (GSVA-IDOC(42)) - guess grid file (upper air) number
17.) NGFS (SRVA-IDOC(29)) - grid file containing 3 surface grids of Z100, TSL and DD
18.) ZGRID (SRVA-IDOC(30)) - grid of Z100 (1000 mb heights)
19.) **TGRID (SRVA-IDOC(31))**— grid of TSL (sea level temperatures)
20.) **DGRID (SRVA-IDOC(32))**— grid of DD (surface dewpoint depressions)
21.) **NO. RETRIEVALS (SRAD/VTPZ-IDOC(100))**— number of retrievals processed
22.) **TYP (IDVA and SRAD-IDOC(50))**— type of retrieval to be done
23.) **GSS (IDVA, SRAD and SRET-IDOC(51))**— type of guess to be used for the retrieval
24.) **SFC (IDVA and SRAD-IDOC(52))**— spacing of attempted retrievals (in fields of view)
25.) **SIZ (IDVA and SRAD-IDOC(53))**— length of one side of a retrieval box in fields of view (retrieval boxes are usually 11*11 fields of view for small detector data, and 5*5 fields of view for large detector data)
26.) **SFC (IDVA, SRAD and SRET-IDOC(54))**— surface option
27.) **ENDL (IDVA-IDOC(55))**— ending line of retrieval area
28.) **ENDE (IDVA-IDOC(56))**— ending element of retrieval area
29.) **BEGL (IDVA-IDOC(57))**— beginning line of retrieval area
30.) **BEGE (IDVA-IDOC((58))**— beginning element of retrieval area
31.) **TER (IDVA, SRAD and SRET-IDOC(59))**— terminal number
32.) **PLT (IDVA, SRAD and SRET-IDOC(60))**— plot option
Subroutines used by LOVA:

1) DOPEN-I
2) DREAD-I
3) TSNIO-I
4) RETIO-I
5) SDEST-I
6) ENKODE-I
7) ENCODE
8) ENCODX
9) ABORT
10) CONTNT
11) LTQ
12) ZECONV
13) MOV
14) CLEANA
15) TQ
16) PRLNX
17) PROOPEN
18) PRPRR
19) LOCK
20) PRWR
21) PRRD
22) UNLOCK
23) PCCL
24) POST
25) BLKA
26) II
27) STC
28) TQMES
29) ITQ
30) MOVOCW
SPVA

SPVA enters keyed in values into either the VASTEXT file or the TOVS documentation file. Specifically for VAS, it can be used to set (or change) MD and gridfile pointers, number of retrievals, grid numbers within the surface gridfile, etc. for VAS retrieval processing. A list of the quantities that can be manipulated is given at the end of this discussion.

First, the VASTEXT file is opened and read via subroutines DOPEN and DREAD. Then, the NW and SE latitude/longitude coordinates of the image area being processed are set or changed in the VASTEXT file (IF (LLNW.NE.0) IDOC(25)=LLNW,etc.). Next, the MD pointers (files or rows) within the VASTEXT file are changed, depending on which keywords and their respective values have been entered in the original keyin (IDOC(35+M)=MESS). Following this, the VASTEXT file words corresponding to the surface data gridfile and its three associated grids (NGFS, ZGRID, TGRID and DGRID) are changed.

In the next stage of the program, the pointer for the first guess gridfile (NGFG) is checked and changed if necessary, and the results up to this point are written into the VASTEXT file via subroutine DWRITE.

In the final step, the keyword parameter for the number of retrievals is checked. If nothing has been keyed in for this parameter (NRET=-1), SPVA is done. Otherwise, the retrieval MD file row header is accessed, which contains the old value of NRET. Then, the respective words of the arrays IRET and IDOC are updated with the new keyed-in value of NRET, after which a call
to subroutine VRTIO (CALL VRTIO(IRET,0,1)) writes the updated row header back into row MDRR of the retrieval MD file, and the updated IDOC array back into the VASTEXT file. At this point, SPVA is finished.

List of quantities that can be set by SPVA:

1.) NRET- number of retrievals
2.) MDNS- MD file for surface data
3.) MDRS- row of surface data MD file
4.) MDNG- MD file for the first guess
5.) MDRG- row of first guess MD file
6.) MDNR- MD file for the VAS retrievals
7.) MDRR- row of VAS retrieval MD file
8.) NGFG- gridfile for the first guess
9.) NGFS- gridfile for surface data
10.) ZGRID- grid within the surface data gridfile that contains Z100 (1000 mb heights)
11.) TGRID- grid within the surface data gridfile that contains TSL (sea level temperatures)
12.) DGRID- grid within the surface data gridfile that contains DD (surface dewpoint depressions)
13.) LLNW- NW latitude/longitude coordinates of area being processed
14.) LLSE- SE latitude/longitude coordinates of area being processed
OPEN AND READ VASTEXT FILE

CHANGE LLNW, LLSE

CHANGE MONS, MORS, MDNG, MORR, MDNR, MDRR

CHANGE NGFS, ZGRID, TGRID, DGRID

CHANGE NGFG

UPDATE VASTEXT FILE

CHANGE NRET

UPDATE BOTH RETRIEVAL MD FILE ROW HEADER AND VASTEXT FILE WITH NEW NRET VALUE
Subroutines used by SPVA:

1) DOPEN-I
2) DREAD-I
3) TSNIO-I
4) DRWRTE-I
5) DCLCLOSE-I
6) VRTIO-I
7) RETIO-I
8) SDEST-I
9) ENCODE
10) ABORT
11) ENCODX
12) CONTNT
13) LTQ
14) ZECONV
15) MOVBN
16) CLEANAN
17) TQ
18) PRLNM
19) PRCLOS
20) PROGEN
21) PRPRPR
22) LOCK
23) PRWR
24) PRRD
25) UNLOCN
26) PRL
27) POST
28) BLKA
29) II
30) STC
31) ITOC
32) MOVNW
33) TQMES

...
GSVA

This program prepares the upper air guess MD file by reformatting the first guess grids in a particular grid file into guess vectors within the guess MD file. Each row and column coordinate within the guess MD file refers to a guess vector at particular latitude and longitude coordinates. Upon completion of the reformatting, the VASTEXT file is updated to show both the original guess gridfile and the resulting guess MD file numbers.

The first major step is to read the first guess gridfile (NGFG) and then open and read the VASTEXT file. Then, the first grid to be reformatted into the guess MD file, as well as the guess MD file itself, are placed into variables NGB and MDNG. Next, after such quantities as the NW and SE guess MD file boundaries (VASTEXT defaults), lat/lon increment and debug option are set, but before the guess grids themselves are read in DO LOOP 10, function MDKEYS returns the keys for the guess MD file. This step is important because it allows GSVA to establish a relationship between grid and MD file scaling (variable SCL), which is calculated later in the program. Scaling relates to the way variables are stored in a given file. For instance, if temperature is stored as degrees Kelvin * 100 in the guess gridfile, but it is to be stored as degrees Kelvin * 1000 in the guess MD file, then the relationship established via SCL will assure that the correct data is being stored, at least in terms of order of magnitude, in the guess MD file.

Following the reading of the keys, the guess grids are read from guess gridfile NGFG in DO LOOP 10 via function IGGET, one at
a time, and ordered for the guess MD file. Variable IOK will
equal 0 for each grid read correctly. As many as 22 grids can
be accessed from the guess gridfile. Among these 22 grids are 15
grids of temperature from 1000–300 mb, 6 grids of dewpoint
temperature from 1000–300 mb, and a grid of 1000 mb heights. The
ordered grids are placed in the array IGRID
(IGRID(J,L)=IGRID(J,23)), with each grid occupying one column of
this array. Upon completion of this task, if the debug option is
on (KBUG .NE. 0), information concerning the day, hour (time of
model initialization), grid level, etc. is displayed on the CRT
via subroutine OUTINT. Note that GSA will exit DO LOOP 10 if
variable NGOT is set to 22 at any point during the execution of
the loop (IF (NGOT.EQ.22) GO TO 12).

After all the guess grids have been accessed from the guess
gridfile, three messages are displayed via the three calls to
subroutine SDEST immediately after DO LOOP 10. Then, the lat/lon
bounds accessed previously from the VASTEXT file (or explicitly
keyed in via positional parameters 4 and 5) are used to generate
latitude/longitude extents (variables LAN, LAS, LOW and LOE) for
the guess MD file. These extents are limited (obviously) to the
extents of the guess grids (see DO LOOPs 50 and 40). DO LOOPs 50
and 40 make up the vector of 26 pieces of guess information (day,
hour (of model initialization), latitude, longitude, 15
temperatures (T), 6 dewpoints (TD) and the 1000 mb height (Z100))
for the guess MD file at a particular row and column (latitude
and longitude) coordinate. The actual loading of the data into
the guess MD file is accomplished by function MDPUT.
(I0K=MDPUT(MDNG,NN,MM,IREC)).

The data is reformatted row by row. In other words, starting in the upper left corner of the guess area, data is loaded in increments toward the right, and when that row is finished, data is loaded from the far left in the second row. In addition, if one needs a first guess which lies between two available guess grid times, a time interpolation option updates the earlier 22 grid values of T,TD and Z100 to the correct time by using both time sets of grid values and an interpolation factor (DT). (FINC=DT*FLOAT(JREC(K)-IREC(K)); IREC(K)=IREC(K)+FINC).

Finally, the guess MD file is closed, and the VASTEXT file is updated with the day, a quantity combining hour and valid time, guess MD file and row numbers, and the guess gridfile number, via subroutine DWRITE. A subsequent execution of program LOVA will show that both the guess gridfile and guess MD file numbers have been filled in the VASTEXT file (MDRG never set, so takes on a value of 0 by default).
Do-Loop 10: Read in and order (guess grids (up to 22 of them)) from guess grid file.

Implicit loop over two time periods.

If NBEF = 0

If L = 0

N = NGB
N = N + 1

N = NGE

A

Following necessary because of grids with holes.

ITYNAM = IGHDT(7)

ITYLEV = IGHDT(10)

I0K = IGHET(NGFG, M, 32, 1, IGRID(1, 23), NR, NC, IGHDT)

Call SODET ('UNABLE TO OPEN GRID FILE NO. ', NGFG)

Order the grids for profile.

If I = 22

Go to 10

A

If 10K < -1

Go to 3

ORDER THE GRIDS FOR PROFILE

If I = 22

Go to 5

A

If IGHET(J) = 1

Go to 8

A

If J = J + 1

Go to 5

B

If J0K = J0 + 1

Go to 8

A

If L = J

Go to 8

A
LOAD GUESS GRID
VALUES OF T, TD OR
2100 INTO ARRAY "GRID"

CALL OUTINT(KOUT)

SET UP FACTOR BETWEEN
GRID SCALING AND
HD SCALING

IGOT(L) = 1

NGOT = 0

J = 1
J <= 22
J = J + 1

NGOT = NGOT + IGOT(J)

12

NGOT = 22

A

950

CALL SDEST ('CANNOT FIND
ONE SINGLE LOUSY GRID', 0)

END

CALL SDEST ('PREPARING
GUESS FOR DAY', IGDAY)

CALL SDEST ('HOUR =
', IGTIME)

CALL SDEST ('VALID
TIME =
', IGTIMEV)

GET DATE INFORMATION
FROM GRID AND MOVE
TO RAW HEADER

NLAT = LAN + IAN

NBUG = 45

SET MD FILE EXTENTS
PRINT GRID AND MD FILE EXTENTS

CALL SDEST('IGLAMX IGLOM, IGLOMN IGLOMN IGLOMN', 
2)

CALL OUTINT(KOUT)

CALL OUTINT(KOUT)

CALL SDEST('MLA = ', MLA)

CALL SDEST('NLO = ', NLO)

N = N + 1
N = NLA

50

E

/IGLAMX, IGLOM, IGLOMN, IGLOMN ARE DETERMINED
BY EQUIVALENCE STATEMENT
AND ARE EQUAL TO IGHD(39) THOUGH
IGHD(38), RESPECTIVELY

MAKE SURE LAT (MD FILE COORDINATE)
IS WITHIN GRID EXTENT

F

NLAT > IGLOM

50

E

F

NLAT < IGLOM

WRITE ROW HEADER

NN = NN + 1

IOK = MDOUT(MDNG, NN, O, IREC)

970

IOK < O

CALL SDEST('CANNOT WRITE ROW HEADER', N)

END

LON = LOW + 10000

M = M + 1
M = NLA

40

F
MAKE SURE LON (MO FILE COORDINATE) IS WITHIN GRID EXTENT

WRITE IS VALUES OF T, 6 VALUES OF TO AND 1 VALUE OF 2166 INTO ARRAY "IREC" FOR ONE PARTICULAR MO FILE LOCATION

DO TIME INTERPOLATION BETWEEN TWO GRIDS (IF DESIRED)

STORE GUESS VECTOR IN GUESS MO FILE (MONG)

F

LON = LON - JINC

END

F

NLAT = NLAT - JINC

CALL SDST ("CANNOT WRITE DATA VECTOR ": L)
Read in first grid of earlier set for time interpolation (NG8), as well as fractional time interpolation factor (DT)

Update array "IDOC" with guess no file numbers, as well as the guess gridfile number, and then write into VASTEXT file
Subroutines used by GSVA:

1) DOPEN-I
2) DREAD-I
3) TSNI0-I
4) OUTINT-I
5) SDESE-I
6) MDCLOS-I
7) DWRIE-I
8) DCLOSE-I
9) ENCODE
10) ENCODX
11) AORT
12) CONTNT
13) ZECONV
14) LTO
15) MOVB
16) CLEANA
17) TQ
18) PRLINK
19) PRCL0S
20) PROPEN
21) PRPRP
22) LOCK
23) PRW
24) PRRD
25) UNLOCK
26) PRCL
27) POST
28) BLKA
29) II
30) STC
31) LWCOLS
32) MOVWC
33) LWMTP
34) MOVCM
35) ITOC
36) LWPO
37) EDEST
38) MOVW
39) CLEANW
40) LWNEWF
41) LWSD
42) JMBWTF
CSVA

CSVA takes surface data from a given MD file, calculates surface dewpoint depression, sea level temperature and 1000 mb height for each station location, and places the resulting data into a surface MD file (MDNS).

After the output MD file (MDO) is opened for read/write and the existing documentation (VASTEXT file) accessed, the status word (IDOC(28)) is checked (was previously set to 1 by IDVA). Then, after the array IDOC is updated with the output MD file number, the MD file from which the conventional surface data is to be taken for the calculations is determined (NFIL=IKWP('MDF',1,0)). If the input MD file is not specified, it will be determined from variable IDAY, and can be either an MD file very close in time to the VAS retrievals being processed, or a file from an entirely different day. After the input MD file has been determined, it is opened for reading, the keys for the file are read via function MDKEYS, and default latitude and longitude limits are set up via IDOC(25) and (26) (LLNW=IDOC(25); LLSE=IDOC(26)). These limits are used to define the extent from which the raw input MD file observations can be taken. Note that the default limits can be overridden by keyed-in values via keyword parameters LAT and LON. Next, the correct row of the input data MD file for the date and time desired is accessed (note that if keyword parameter AUTO is used, the first row of the correct day will be picked), and a row header for row ITIME of output MD file MDO is written. Then, the program begins gathering station data in implicit DO LOOP 40 from row ISV of
input MD file NFIL using function MDGET
(IIK=MDGET(NFIL,ISV,NREP,IBUF)), making sure the station is
within the latitude-longitude bounds set earlier.

For each station whose data is read, the station dewpoint
depression, sea level temperature, and 1000 mb height are
calculated. Note that a missing station temperature value will
result in none of these quantities being calculated for a given
report, and that there must be station temperature and dewpoint
values, as well as a sea level pressure value, for all three of
the quantities to be determined.

The results for each station (the array KBUF) are placed in
row ITIME of output MD file MDO via function MDPUT
(IIK=MDPUT(MDO,ITIME,NOUT,KBUF)), and the program goes back to
statement 40 to get another station's data. Note that provisions
are made in case there are not enough station reports accessed
(variable NOUT after code line "50 CONTINUE"). CSVA will try to
get surface data from 1, or possibly 2 hours earlier than ITIME,
if necessary. Then, if it does not find at least 10 surface
reports for either earlier time, a message saying "INSUFFICIENT
SURFACE DATA AVAILABLE.." will be printed on the CRT, and the
program will transfer to statement 53.

Finally, assuming all available data is accessed with no
problems, the number of reports written (statement 53) is
outputted to the CRT, and the array IDOC is updated and written
into the WASTEXT file (CALL DWRITE(LUN,ITERM,IDOC)). In addition,
the row header of the output MD file is re-written to show the
number of surface data reports written into that row of the file.
A subsequent execution of program LOVA will now show the retrieval surface MD file number, plus the row of that file which contains the station dewpoint depressions, sea level temperatures and 1000 mb heights for all the stations accessed.
Determine input MO file

NFIL = IKWP('MDP', 1, B)

11

Assume file is in current section

NFIL = MOD(IDAY, 10)

IKWP (SHIFT, 1, 0) ≠ ∅

NFIL = NFIL + 10

NFIL = ∅, NFIL = NFIL + 10

KBUG ≠ ∅

CALL SDEST('OPENING HOUR NO. ', IHR)

OPEN input MO file for read

OPEN = MDOPEN(NFIL, 1)

CALL SDEST('UNABLE TO OPEN INPUT FILE ...

END

970

END

Read keys of input MO file

IKEY = MDKEYS(NFIL, LIST, ISCL, UNITS, LOC)

12

CALL OUTINT(KOUT)

CALL OUTINT(KOUT)

CALL OUTINT(KOUT)
Subroutines used by CSVA:

1) HRTopo-I
2) DOPEN-I
3) DREAD-I
4) TSNIO-I
5) OUTINT-I
6) SDEST-I
7) DWRITE-I
8) ENCODE
9) ABORT
10) ENCODX
11) CONTNT
12) LTQ
13) ZECONV
14) MOVX
15) CLEANX
16) TQ
17) PRLINX
18) PRCLOSE
19) PROPEM
20) PRPRPR
21) LOCK
22) PRWR
23) PRPD
24) UNLOCK
25) PRCL
26) POST
27) BLKA
28) III
29) STC
30) DCLOSE
31) ITOC
32) MOVCEW
SRVA

The purpose of SRVA is to generate grids of surface variables previously calculated by program CSVA for use in either VAS or TOVS retrievals. As usual, only VAS will be dealt with here. The output of SRVA is stored in a grid which, in turn, is contained within a gridfile for surface data. Normally, the user will run SRVA three times to generate one grid each of sea level temperature, surface dewpoint depression and 1000 mb height.

First, after the VASTEXT file has been opened and read via subroutines DOPEN and DREAD, variables such as retrieval surface MD file and row (MDNS and MDRS, respectively), guess MD file (MDNG), etc. are set. Then, the surface MD file prepared by CSVA (MDNS) is opened for read/write, and the number of reports to be processed (NRPT) is determined, either by reading the row header for row MDRS or an explicit keyin of keyword parameter RPT. These reports will contain the data that is to be used in constructing the surface grid.

After the operator-forced boundaries (set up in IDVA or keyed in) that will determine the area of the grid are determined, the character whose data is to be gridded is set by function CPP (CHAR=CPP(1, ' ')), and the analysis field arrays IDL and ITG are initialized to 0. Note that several different types of variables can be gridded (TSL (sea level temperature), DD (surface dewpoint depression), TS (surface temperature) and Z100 (1000 mb height)). In addition, note the element of the gross error check array (SDL) that is set up after the character to be gridded has been determined. This element is important later in
both a gross error check involving the guess MD file (MDNG) guess field (compares raw data value to guess value) and within an edit option quality control check (compares raw data value to final grid value). After the IDL and ITG arrays have been initialized to zero, the grid increment (in tenths of degrees) is established. Keep in mind that this increment can be either keyed in or determined within the program. If the grid increment is NOT keyed in, it will be calculated such that the total number of gridpoints is less than or equal to 2400 (the value of variable NSIZE).

At this point, it is possible to acquire the information to be used in the surface analyses in two different ways. First, if a guess is to be used when generating the grids, such as will be the case if there are large data-void areas within the analysis region, a call to subroutine SRGSS will access the guess (the "VASGSS" route), after which the actual surface data will be gathered via function MDGET (beginning of DO LOOP 148). It should be mentioned here that the guess grid option is never used when executing SRVA and can therefore be ignored within the context of this discussion. Note the "NODAT" option approximately 15-20 lines after the call to subroutine SRGSS. If it is in effect (IKWP('NODAT',1,0) .NE. 0), SRVA will skip to statement 150, and guess data only will be stored in output surface gridfile NGRFS. In addition, note the preparation of variable FCK just before SRVA enters DO LOOP 148. This variable is used later when comparing the raw surface data to a value from the guess field to determine if a given data value should be used in the
Barnes analysis.

On the other hand, if no guess is desired (no keyin for keyword parameter GSS, or variable CGS = ')'), SRVA simply moves to DO LOOP 148 to read the actual surface data. This option should normally be used in the event there is good data coverage.

SRVA now enters DO LOOP 148. This loop reads each individual surface report (which contains the 1000 mb height, sea level temperature, dewpoint depression, etc. data) from the surface MD file (MDNS). In addition, it makes sure a given report is within the previously-set LAT/LON bounds, performs a test (only for the "VAGSS" guess option) to screen out very poor surface data, and fills Barnes arrays such as DAS, RW and CL for use in the forthcoming Barnes analysis. Upon completion of DO LOOP 148, the Barnes analysis is set up and executed. The analysis itself is carried out by subroutine FBARN, and results in a uniform latitude-longitude grid of values.

At this point in SRVA, assuming the edit option is on (IKWP('EDIT',1,0).EQ.0), the standard deviation of the actual data used in the grid generation (FDAT) is computed and stored in variable SD. Then, if the difference between a given raw data value and a value derived from the final grid (VAL) is .GE. to the standard deviation, the original surface data report is flagged with variable MISS, and the program returns to statement 110 to read in the surface data again and repeat the Barnes analysis. This time, however, the flagged report(s) will not be taken into account in the Barnes analysis.

Assuming there are no problems with the Barnes analysis grid
and the edit option passes, an output surface gridfile (NGRFS) is opened (IOK=IGOPEN(NGRFS,IFIL2)), and the sea level temperature, dewpoint depression, 1000 mb height, etc. grid is placed within it by function IGPUT. In addition, the array IDOC is updated to show both the output surface gridfile number and the number of the new grid within this gridfile (ISTAT). This information is then subsequently written into the VASTEXT file via subroutine DWRITE.
IF THERE IS NO SFC, MO FILE DATA FOR USE IN THE GRID GENERATION AND A WEAK MO FILE GUESS IS ALREADY USED, JUMP TO STATEMENT 150 AND:
CALL GSOE (BEGIN BARNES ANALYSIS WITH SCALING OF Z, 1ZCAL)

PERFORM BARNES ANALYSIS TO GENERATE UNIFORM GRID OF GIVEN VALUES

CALL FBNR(TLAT, NWNS, NCOLS, FLID, W1, W2, D, RN, C, N, NUM, J, JCAL, 168)

AVOID NEGATIVE DEWNPS WEEP DEPRESSION
CHECK ON THE EDIT OPTION

IF EDIT OPTION NOT IN EFFECT, GO TO STATEMENT 150 AND SUBSEQUENTLY STORE THE FINAL GRIDDED SURFACE ANALYSIS IN GRID FILE "URPS"

FCK = .999999

CALCULATE STANDARD DEVIATION OF INPUT SURFACE MO FILE DATA

SD = DSQRT(SUMSUM-SUMSUM)

KBUG = 15

CALL GSOE ('DATA SD 15', 1SD)

LIMIT SIZE OF STANDARD DEVIATION

MV+1 = MV+1
NN+N1 = NN+N1

GET VALUE FROM GRID

GET VALUE FROM FINAL ANALYZED GRID

CALL VALUE (LAN, LAS, LOW, LOE, ENC, JDL, 184, VAL)

VAL = .999999

GROSS ERROR CHECK

ADIF = ABS(DIF)

COMPARE DIFFERENCE BETWEEN FINAL GRID VALUE AND SFC, MO FILE VALUE WITH THE STANDARD DEVIATION COMPUTED EARLIER; IF THE RESULT INDICATES THAT THE MO FILE VALUE WAS BAD, GET THE REPORT CONTAINING THE FAILURES FROM THE SFC AND UPDATE THE VALUE TO "MISS" AND PUT THE UPDATED REPORT BACK INTO THE SFC, MO FILE, THEN GO BACK TO STATEMENT 150 AND REPEAT THE BARNES ANALYSIS.
CALL ENQ5E (YIX, IS, ZFI, ZS, 
2F91, 1), LE, NW, FLAT, 
FLON, DAT, VAL)

IF POOR DATA WAS REMOVED FROM 
ANY SFC. ND FILE, REPORTED TO
TO STATEMENT
110 * READ

STATEMENT 150

OK = IGOPEN(NGAFS, FILE)

CALL SDEST(UNABLE TO OPEN GRID FILE..; φ)

OK = IGPUT(NGAFS, NGAM, IDL, NWMS, NCOLS, IDPC, ISTAT)

UPDATE VASTEXT FILE WITH 
SURFACE GRIFFILE *(STORED 
IN IDOC(00)), AS WELL AS *OF 
GAIO JUST PUT INTO SFC. GRIDFILE 
(STORED IN IDOC(28 + LA))
CLOSE VASTEXT FILE

CALL INFO(3, 1, 1, 1, 
IINOC)

END
Subroutines used by SRVA:

1) SDEST-I
2) VALUE-I
3) DOPEN-I
4) ENKODE-I
5) DREAD-I
6) TSNIO-I
7) SRGSS-I
8) OUTINT-I
9) FBARN-I
10) TQMES-I
11) DWRITE-I
12) DCLOSE-I
13) BLKA
14) MOVESW
15) MOVB
16) LTO
17) CLEANA
18) TQ
19) PRLNX
20) PROCLOS
21) PROPEN
22) PRPRPR
23) LOCK
24) PRWR
25) PRD
26) UNLOCK
27) PRC
28) POST
29) ABORT
30) ENCODE
31) ENCODX
32) CONTNT
33) ZECVNV
34) II
35) STC
36) ITOC
37) RASGES
38) FILL
39) LINFL
40) INTER
41) HEPFY
XRVA

XRVA is the main editing program used in conjunction with VAS retrieval processing. It can be used to delete entire report(s) contained within the cursor, a parameter at a given level in all report(s) contained within the cursor, or a parameter at all levels in all report(s) contained within the cursor. (As a side note, since the cursor size can be varied by using McIDAS program "DF," it is possible to cover either one report or several reports at one time under the cursor.) If the user is working on a surface data MD file, he/she can delete either an entire report or reports, or only a particular parameter value for single or multiple reports, within the cursor. The parameter deleted can be any valid variable which exists in the schema for that particular surface MD file, but it will usually be one of Z100 (1000 mb height), TSL (sea level temperature) or DD (surface dewpoint depression). On the other hand, if the user is working on a retrieval MD file, an entire retrieval can be deleted if keyword parameter NRET is keyed in as the only positional or keyword parameter, or one or more entire retrievals contained within the cursor can be deleted if XRVA is keyed in by itself. In addition, a particular parameter at a keyed-in level for a retrieval or retrievals can be deleted, or a particular parameter at all levels for a retrieval or retrievals can be deleted. Note finally that one must do an "MDU SET" before the execution of XRVA to assure that McIDAS will be operating on the correct surface or retrieval MD file.

After the MD file number listed in User Common is read via
function LUC (MDNO=LUC(5)), the VASTEXT file is opened and read, and the cursor line/element position (IL,IE) is determined via subroutine TVSAT. Then, the navigation is initialized, and the MD file whose number was determined earlier is opened for read/write, provided that its number matches either IDOC(36) or (40) (surface and retrieval MD files, respectively, as defined in the VASTEXT file).

Just before statement 2, XRVA checks to see if an entire retrieval with the specified number should be deleted. If this is the case (NRET .NE. 0), the report which exists in row MDR, column NRET of MD file MDNO will have all of its values changed to MISS, after which XRVA will terminate. However, if this is not the case, (NRET .EQ. 0), XRVA jumps to statement 99, where the row header of MD file MDNO is read, which gives the number of retrievals/surface reports in that particular row (variable MREC). At this point, a test to check the MD file the user wishes to edit is performed to see if data exists ("READ TEST RECORD"). Assuming there is no problem with this test, the keys corresponding to the schema for that MD file are read by function MDKEYS. Then, after image information has been accessed (GETFRM) and the plotting routines initialized (INITPL), XRVA checks to see what parameter should be deleted (variable ICHR), and what level (or all levels, if no level is keyed in for the second positional parameter) it should be deleted from (variable ILEV).

If no parameter or level are specified, control passes directly to statement 10, and the process to delete ALL reports at ALL levels under the cursor begins. However, if some parameter
(such as Z, T, etc.) is keyed in as the first positional parameter, but nothing is entered as the second positional parameter. DO LOOP 3 sets the array IADD and the variable KTOT, both of which will be used later (in DO LOOP 58) to delete that parameter from all levels in each report which is contained by the cursor. Finally, if both a parameter AND level are keyed in, DO LOOPs 5 and 7 prepare IADD and KTOT such that the given parameter will be deleted at that level from all reports within the cursor.

At this point, XRVA is in the vicinity of statement 10. After the "GO" prompt is displayed on the CRT by function ICURG, the user should place the cursor over the report(s) to be edited and then push the spacebar, which will result in an edit box for the forthcoming deletion(s) being traced out around the outside dimensions of the cursor by subroutine WRBOX. Then, the latitude/longitude coordinates for the four corners of the cursor box are stored in the arrays FLA and FLO, and another "GO" appears on the CRT. The user should then position the cursor over the next report(s) to be edited and push the spacebar again. As few as 1, or as many as 40 edit boxes can be traced out on the video screen (exact number of boxes stored in the variable NX).

The deletion stage of the program begins either when NX=40 or something other than the terminal's spacebar is punched after the "GO" prompt. Punching a character other than the spacebar gives function ICURG a value of 1. Usually, "R END" will be typed by the user to terminate the box-drawing stage of XRVA.

In the next step, since LUC(81) (the VAS sounder area
number) is .NE. 0, XRVA enters DO LOOP 70, which carries out the actual deletions. Initially, a given MD file report is read (IOK=MDGET(MDNO,MDR,M,IOUT)) and, provided the report has not been previously marked for deletion via the "mod" flag (NMOD), the program enters DO LOOP 40. Within this loop, the latitude/longitude coordinates of a given MD file entry (retrieval or surface report) are compared with the maximum/minimum latitude/longitude coordinates of each of the NX plotted edit boxes. If a given MD file report is found to be contained within one of the edit boxes, the raster and pixel (TV (television) coordinates) of the retrieval or surface report are calculated via subroutines SATEAR and SATTV and stored in the variables IRAS and IPIC. Then, assuming the MD file report also is within the raster/pixel bounds of the edit box (IRMIN, IRMAX, IPMIN, IPMAX), a red "X" is plotted over the report location by subroutine WRMAR.

Then, if a given parameter at a level or all levels is to be removed from the report (ICHR.NE.' '), the element(s) of the array IOU T corresponding to the data to be deleted is(are) set to the value MISS in DO LOOP 58, after which the updated array is put back into the surface or retrieval MD file via function MDPUT (IOK=MDPUT(MDNO,MDR,M, IOUT)). Following this, XRVA loops back to the beginning of DO LOOP 70 to process the next MD file report.

On the other hand, if the entire MD file report is to be deleted (ICHR.EQ. ' '), the element of IOUT which marks the report for future exemption (the mod flag) is set to the value
999999, after which the updated array is returned to the surface or retrieval MD file via MDPUT.

Finally, DO LOOP 70 concludes, the VASTEXT file is closed, and a message saying "ALL DONE DELETING ..." is printed on the CRT.

As a final note, a few sample keyins which relate to the previous text will be presented. These examples should help settle any confusion concerning the previous discussion.

1.) XRVA: Delete entire report(s) that lie within the cursor from MD file whose number was set before XRVA via the "MDU SET."

2.) XRVA Z100: Delete all values of 1000 mb height from report(s) located within cursor, with report(s) again being taken from MD file whose number was set via program MDU.

3.) XRVA NRET=105: Delete report #105 (exists in column 105) from row MDR of MD file whose number was set by MDU.

4.) XRVA Z 500: Delete all values of 500 mb temperature from report(s) contained within the cursor (report(s) once again taken from MDU-defined MD file).
IF "NRET" ≠ Ø, DELETE ENTIRE "NRET" RETRIEVAL

DETERMINE TOTAL NUMBER OF MD FILE REPORTS THAT CAN POTENTIALLY BE EDITED

READ KEYS FOR MD FILE

NKEYS = MDKEYS (MDNO, -1, list, ISCL, IUN, LOCs)
GET IMAGE INFORMATION; INITIALIZE PLOT ROUTINES

DETERMINE CHARACTER TO BE EDITED

DETERMINE LEVEL FOR EDITING

CODE FROM HERE TO STATEMENT 10 DETERMINES WHICH DATA VALUES IN GIVEN REPORT WILL BE DELETED (EDITED)
Determine max/min latitude/longitude for given plotted editing box.
CONVERT FROM SATELLITE TO TV (TELEVISION) COORDINATES

PLOT RED "X" OVER EDITED REPORT

CARRY OUT DELETIONS

END
Subroutines used by XRVA:

1) DOPEN-I
2) DREAD-I
3) TVSAT-I
4) NVINIT-I
5) TSNIO-I
6) SDEST-I
7) GETFRM-I
8) INITPL-I
9) WRBOX-I
10) FLOC-I
11) OUTINT-I
12) SATEAR-I
13) ENDPPLT-I
14) SRCH-I
15) SATT-I
16) WRMAR-I
17) DCLOSE-I
18) ENCODE
19) ENCODX
20) ABORT
21) CONTNT
22) LTQ
23) ZECONV
24) MOV
25) CLEAN
26) TQ
27) PRLNX
28) PRCLOS
29) PROPEN
30) PRPRPR
31) LOCK
32) PRWR
33) PRRD
34) UNLOCK
35) PRL
36) POST
37) BLKA
38) TMNLI
39) TQMES
40) STC
41) ITOC
42) GETNAV
43) EPOCH
44) GETGAM
45) VASNAV
46) DDEST
47) MOVW
48) CLEANW
49) EDEST
50) MOV
51) III
52) DWRITE
VTPZ

VTPZ is the operational VAS retrieval program. For a given retrieval, it generates vertical profiles of temperature and moisture by using a "one-step" physical matrix inversion retrieval algorithm (Smith and Woolf, 1984). VTPZ can be run to generate either one retrieval or a larger area of retrievals, depending on how the program is keyed in on McIDAS. In addition, an option exists to create an image of total precipitable water vapor and also possibly another image of total-totals index simultaneously with the retrievals. These image(s) are stored in separate McIDAS digital areas (of type VISSR). Finally, VTPZ can be run in either of two modes: auto (non-video) or non-auto (video). If the user doesn't have access to a video terminal, the program must be run in the auto mode, while video terminal users can run VTPZ in either of the two modes. In this discussion, the operation of VTPZ will be presented from the standpoint of the non-auto mode. Pertinent differences between this mode and the auto mode will be duly noted as the discussion proceeds.

Initially, after a message saying "BEGIN*VTPZ*VERSION OF 3 MAY 84 AT INIT 2" is displayed on the CRT, the number(s) of the digital area(s) into which image(s) of total precipitable water vapor and total-totals index is(are) to be placed is(are) set. Note that an image of total-totals index cannot be created independent of an image of total precipitable water vapor; that is, the user can either create both images, or one image of total precipitable water vapor. This is to be understood throughout the remainder of this discussion. (As a side note, if the user wishes
to know the correct number of retrievals made in the retrieval area, he/she should, before executing VTPZ, execute program SPVA as follows: SPVA NRET=0.) After the number of areas to be generated (0.LE. NAREAS .LE. 2) has been calculated, a call to subroutine VRTIO is encountered. This subroutine opens, reads and closes the VASTEXT file, opens the retrieval MD file (MDNR) for read/write, and reads the row header for the same file. The information contained in the VASTEXT file (more explicitly, in the array IDOC) allows VTPZ to set variables JDAY, JTIME, MDNG, MDNR, and MDRR. Note that MDNG, MDNR and MDRR are, respectively, the retrieval guess MD file number, retrieval MD file number, and retrieval MD file row number.

Now, assuming the retrieval MD file row number is .GT. 0 (IF(NARA.EQ.0.AND.MDRR.LE.0) GO TO 1300), the type of guess to be used for the retrievals ("G"= grid (default), or "C"= climatology) is evaluated, and variable IGES is set accordingly to 0 or 1. After the surface option is set, the size (in fields of view (FOVs)) of the box in which retrievals are to be made (NBXS) is determined. If the user does not key in any information about the retrieval box size, it will become 11 FOVs in dimension by default, and variable LDETR will assume a value of 0. However, if the user desires a different retrieval box size, and the VAS data was gathered via the large detectors (resolution = approximately 16 km), keyword parameter BOX should be keyed in with an EVEN integer value ("BOX=6," for example). Assuming BOX is keyed in as 6, the resultant retrieval box size will be 5 X 5 FOVs (the typical box size for retrievals using large-detector

117
VAS data). In addition, variable LDETR will be set to 1. On the other hand, if the data was gathered via the small detectors (resolution = approximately 8 km), the retrieval box size should be keyed in as an ODD integer value ("BOX=9," for example). With this sample keyin, the resultant retrieval box size will be 9X9 FOVs (NBXS-MINO(NBXS,11)), and LDETR will be set to 0. Note that the range of retrieval box sizes possible with VTPZ is 1 to 11.

Subsequent code determines the retrieval spacing in FOVs in both the line (y) and element (x) directions (INCRL,INCRE), and sets the end/beginning coordinates of the retrieval area to be processed. Note the VASTEXT defaults for the end/beginning coordinates. After the cursor location in TV coordinates (rasters, pixels) has been evaluated (INRAS=LUC(-11), INPIC=LUC(-12)), variable CAMRET (used in retrieval subroutine QVTWR) is determined, and the system date is derived via subroutine GETDAY. At this point, the dimensions of the area (or coordinates of a single retrieval) to be processed are defined. There are numerous different methods of defining areas to be processed, and at least two ways to process only one retrieval. However, the usual case is to key in the ending line/element coordinates of the retrieval area, and let the VASTEXT file default values determine the beginning line/element coordinates (see sample keyin of VTPZ for the non-auto mode within instruction 22 of the sample keyins near the end of chapter 1). Note that a keyin of this type will result in a call to subroutine GETFRM (returns a frame directory relating to the sounder area previously loaded into a given image frame), while subroutine TVSAT will not be called. If the
user was running VTPZ in the auto mode on either a video or non-video terminal, there would be no call either to subroutine GETFRM or TVSAT. (Note: LUC(16) = 1 for a video terminal and 0 for a non-video terminal.) Again, see the sample keyin of VTPZ, this time for the auto mode, at the same location in Chapter 1.

Following the setting up of the area to be processed, subroutine VASDAT is called. Since VDAT(1)=-1, when the routine is called, only navigation information is returned. Included in this navigation information is satellite number, Julian date, time the first VAS data in the image was gathered, and line and element resolutions of the image. Next, variables INCIL and INCIE are calculated. INCIL stores the number of satellite coordinate lines separating successive retrieval lines, while INCIE contains the number of satellite coordinate elements separating successive retrieval boxes on any given retrieval line. These two variables are used subsequently to increment the two most important DO LOOPS of the program -- 1260 and 1180. The last lines of code before DO LOOP 190 set several more variables, most notably variables K.Lines, KELEMS and KBOXES. The first two deal with retrieval box dimensions, and both have the same magnitude as variable NBXS discussed earlier. KBOXES, on the other hand, holds a value indicating how many retrieval boxes can fit in the x-direction of the area to be processed; that is, on a given retrieval line. For a visual summary of the previous discussion, see Diagram 1 immediately preceding VTPZ's Level I flowchart. This diagram shows approximately how a small (only 4 retrievals wide) retrieval area looks once it is set up for
processing.

Assuming the user wishes to generate at least an image of total precipitable water vapor (NARA .NE. 0), and assuming also that MELE (number of elements in the x-direction of the area to be processed) is not .GT. 660, DO LOOP 190 is now entered. This loop creates and prepares for data insertion up to two digital areas, one to store the image of total precipitable water vapor, the other (if desired) to store an image of total-totals index.

At this point, after a message concerning the opening up of the precipitable water vapor digital area is displayed on the CRT, VTPZ enters DO LOOP 1260. This loop increments each time a given line of retrieval boxes has been completely processed. Note also the beginning, ending and increment values of the loop (ILINE, LLINE and INCIL). Almost immediately after DO LOOP 1260 has been entered, DO LOOP 1180 is encountered. This loop, nested within 1260, increments each time a retrieval box within a given retrieval line has been completely processed.

The first thing done in DO LOOP 1180 is an initialization of the arrays TDAT and IARRAY. TDAT will eventually hold the measured VAS brightness temperature (TBB) data for all 12 bands for each FOV in a given retrieval box being processed. On the other hand, IARRAY, which is used only in the image-generating section of VTPZ, will hold the total precipitable water vapor amount and (if a second image is desired) the total-totals index for each FOV in all the retrieval boxes in a given retrieval line.

After these arrays have been initialized, TBB and other
miscellaneous data for each FOV of a given retrieval box is accessed in DO LOOPs 330 and 320. The subroutine which does the actual data-gathering is VASDAT. The input arguments (IL, IE) give the line/element coordinates of the FOV for which the data is to be collected, while the array VDAT is used to return the TBB data itself. The other data is returned via common blocks such as DANGLE, LAST and NAV. After the data for a given FOV has been accessed, several data quality tests are performed, including tests for latitude, satellite zenith angle, land elevation, and whether TBB data exists for that particular FOV.

Assuming these tests pass, VTPZ enters DO LOOP 300. This loop, which runs through all 12 bands for a given FOV, begins with a check to delete extreme TBB values for a given band. Assuming the data for a given band is reasonable, it is stored in the array TDAT, and the spin budget for that particular band is saved in the array NSPIN. Then, if the loop is on band 8, variable MSAM (number of FOVs with good data in a given retrieval box), SELEV (sum of surface elevations for all the good-data FOVs in a given retrieval box), and LSTYPE (number of FOVs over land in a given retrieval box) are updated.

Upon the exit of DO LOOPs 330 and 320, and assuming at least one decent FOV of data was found in the retrieval box, the mean surface elevation for the box (SELEV=SELEV/FLOAT(MSAM)) is calculated, and the program proceeds to obtain the first guess for the retrieval.

The subroutine which gathers the first guess is GESPRO. Returned via this subroutine are 40 values of guess temperature
and mixing ratio (at levels from 0.1 to 1000 mb), as well as other assorted surface parameters (such as surface pressure, temperature and dewpoint). After the surface mixing ratio is calculated via subroutine WMIX, the first pressure level below the surface (IS) is determined in DO LOOP 340. This level will be used to define the lowest level in the forthcoming retrieval.

Note that level IS can never be .GT. 40 and can, in rare situations, be equal to the actual surface pressure. Usually, however, the lowest level of the retrieval will not be equal to the actual surface pressure. After IS has been evaluated, the guess temperature profile is adjusted below 700 (ILO=35) or 500 (ILO=31) mb to take into account the surface temperature, 700 mb being the top level of adjustment for a grid first guess (usually the LFM), and 500 mb being the top level of adjustment for a climatology first guess. See the code immediately preceding DO LOOP 380 and DO LOOP 380 itself for details.

Following DO LOOP 380, the temperature and mixing ratio arrays for all the retrieval levels from level IS down to level 40 (1000 mb) are set equal to the value found at level IS. This is done in DO LOOP 420. DO LOOP 440 then adjusts the entire guess moisture (mixing ratio) profile (all 40 levels) to take into account the surface mixing ratio value. Note that the final mixing ratio at a given level is constrained to lie between .02 g/kg and the saturation value, inclusive. As the last step in DO LOOP 440, the final guess temperature and mixing ratio values for all 40 retrieval levels are saved in the arrays TG and WG.

The next major step in VTPZ is the calling of subroutine
FILTER. This subroutine generates retrieval box-averaged TBBs in each band 1-12, as well as the expected error of the TBB observations for each band (array ERM), the number of non cloud-contaminated FOVs for the given retrieval box (MSAM), and a new average retrieval box surface elevation which makes use only of the non cloud-contaminated FOVs (see variable IELEV in common block SURF). Upon returning to VTPZ, a check is made to make sure average TBB data was returned in at least 7 of the first 10 VAS bands (DO LOOP 480). Assuming that this check passes and that there is no problem in calculating the surface skin temperature (TSKIN), variables such as ERTSFC and ERWSFC are set. These variables are used in subroutine QVTWR, which does the actual "one-step" retrievals, to define the expected errors in the input surface temperature and mixing ratio, respectively. Thus, the retrieved surface temperature and mixing ratio are constrained to satisfy the input surface values to within these nominal error estimates. Note also the re-calculation of variables TSTA, WSTA and TDSTA using band 7 and 8 TBB information in the event the surface option is not being used. This is done to give a better first guess at the surface in these cases than would be accessed via climatology. Finally, just before the retrieval is performed, DO LOOPs 1000 and 1020 store guess temperature and dewpoint data, respectively, in the arrays TGS and DGS.

Now, the most important section of VTPZ begins -- the retrieval of the temperature and moisture profiles. The subroutine which does the actual retrieval is QVTWR. For further information about the retrieval algorithm, see Smith and Woolf.
If a given retrieval fails, the error flag IFAIL is set to 1 within QVTWR, and VTPZ will move to the next retrieval box to attempt a retrieval there. Assuming, however, that the retrieval is successful, the resultant temperature and moisture (mixing ratio) output is saved in the arrays TG and WG (DO LOOP 1030), after which the retrieval subroutine is called again. This time, however, variable GAMRET holds a value equal to 1/10 the value it held when the first retrieval was done. In effect, this second retrieval "fine tunes" the result from the first retrieval; that is, the user can think of the first retrieval as a first guess for the second retrieval.

Assuming the second retrieval is successful, VTPZ checks to see if the retrieval box had 5 or more non cloud-contaminated FOVs. If there were less than 5 non cloud-contaminated FOVs, the program jumps to statement 490, meaning that this particular retrieval will be used at most only as a contributor to the image(s) of total precipitable water vapor and total-totals index (if desired), and will NOT be saved in the retrieval MD file. Otherwise, if there WERE 5 or more non cloud-contaminated FOVs, the retrieval is satisfactory for inclusion into the retrieval MD file.

Before the retrieval is stored in the retrieval MD file, however, several additional calculations are made. First, a call to subroutine PRECW results in the calculation of total precipitable water vapor down to the lowest level of the retrieval (IS). Following this, dewpoints are calculated from 300 to 1000 mb, the total-totals index is determined, and the
retrieval profile heights (subroutine HTV) are computed. Next, preparation of the arrays PST, TST and TDST precede a call to subroutine SNDANL, whose function is to compute stability indices (such as lifted, K and SWEAT) for the just-completed retrieval. Finally, if the debug option is on (keyword parameter BUG keyed in with a 1 or 2), information concerning the retrieval profile, as well as the total-totals index value computed earlier, will be printed on the CRT (a keyin of "BUG=1" in the VTPZ keyin) or local printer (a keyin of "BUG=2" in the same).

At this point, VTPZ is ready to begin preparing the output buffer; that is, it begins filling the array IRET with retrieval quantities for insertion into the retrieval MD file. The filling of IRET occupies the program for approximately the next 60 lines of code. The data inserted into IRET consists of things such as the user mod flag (IRETD(3) -- a setting to 0 indicates that a good retrieval was performed), retrieval number (IRETD(4)), as well as the averaged TBBs and expected error of the TBB observations for bands 1-12 (DO LOOP 1140). (It should be mentioned at this point that an EQUIVALENCE statement near the beginning of the program equivalences the arrays IRET and IRETD (EQUIVALENCE(IRET(17),IRETD(1))). Other quantities stored in IRET include the total precipitable water vapor amount (IRETD(35)), and the total totals and lifted indices (IRETD(41) and (43), respectively). Next, DO LOOP 1160 stores the retrieval heights and both the retrieval and guess temperature and dewpoint arrays (only at or above the surface). A final input of some surface parameters completes the retrieval MD file output buffer. After
the total number of retrievals performed so far (LASRET) is incremented by 1 and placed in IDOC(100) (note: the array IDOC holds current VASTEXT file data), a call to subroutine VRTIO (CALL VRTIO(IRET,LASRET,1)) puts IRET into column LASRET (and row MDRR as specified in the VASTEXT file) of the retrieval MD file.

At this point, if no image(s) of total precipitable water vapor and total-totals index are to be generated (NARA .EQ. 0), DO LOOP 1180 increments one retrieval box spacing, and an attempt to process the next retrieval begins (IF(NARA.EQ.0) GO TO 1180). However, if image(s) of these two quantities is(are) desired (NARA .NE. 0), VTPZ proceeds to DO LOOPS 560 and 540. Together, these two loops generate value(s) of total precipitable water vapor (TPWV) and total-totals index for EACH FOV in a given retrieval box (remember that the earlier retrieved values of TPWV and total-totals index were for an ENTIRE retrieval box, not for each individual FOV). The process is carried out in a scan line-by-scan line fashion within each box. The precipitable water vapor and its image value are calculated first (see subroutine PRECW and variables URET/IURET approximately 30 lines into DO LOOP 540). The final vapor amount is limited to the range 1.-12.cm, inclusive, while the vapor image value, stored in IARRAY(KELE,KLIN,KBOX,1), is limited to the range 20-240, also inclusive. (As a side note, an image value of 0 is pure white, while a value of 255 is pure black.) Then, if a second image of total-totals index is desired (IAREA(2) .NE. 0), and the surface pressure is .GE. 850 mb, a value of total-totals index (TOTL) and a corresponding image value (ITOTL) are calculated, with the
image value being stored in IARRAY(KELE,KLIN,KBOX,2). This step then concludes the image-generation section of VTPZ. Note that the program is now at the end of DO LOOP 1180, so control will pass back to the beginning of the loop, where another retrieval will be attempted.

Once a given retrieval line has been completely processed in the retrieval area (DO LOOP 1180 exited), VTPZ checks to see if image (single FOV) total precipitable water vapor and (if desired) total-totals index data was generated previously in DO LOOPs 560 and 540 (IF(NARA.EQ.0) GO TO 1255). Assuming an image of at least one of these quantities is to be outputted to digital area(s) (IF (NARA.EQ.0) GO TO 1255 is NOT true), VTPZ enters DO LOOP 1250. This loop generates one retrieval line image of total precipitable water vapor, and then (if desired) generates a second retrieval line image of total-totals index. Note that the loop processes the data in one retrieval line by completely processing all the FOVs in successive scan lines (DO LOOPs 1220 and 1200), until all the scan lines contained within the retrieval line have been processed.

The data itself is taken from the array IARRAY (recall that the raw data for the image(s) was stored in this array). Then, subroutine PACK stores one scan line's worth of data in the array IABUF, after which subroutine WRITA writes this data into the digital area KARA. At this point, if variable IAGAP (the difference between the retrieval line spacing (in FOVs) and the retrieval box size) is .LE. 0, DO LOOP 1250 is either done (if only a retrieval line image of total precipitable water vapor is
desired), or it loops back to generate a second retrieval line image of total totals index. Assuming IAGAP .GT. 0 (meaning a segment of empty scan lines exists between successive retrieval lines), subroutines PACK and WRITA (DO LOOP 1240) fill in the empty image space in the digital area with default image values (the constant MAXWV, which = 255 (white)).

With the conclusion of DO LOOP 1250, VTPZ has completely processed one line of retrieval boxes within the retrieval area. DO LOOP 1260 now loops back to its beginning, incrementing the retrieval line coordinate in the process by the amount INCIL (which will then give the initial line coordinate of the sequence of retrieval boxes to be processed in the next retrieval line).

If all the retrieval lines within the retrieval area have been completely processed, DO LOOP 1260 exits. Then, the arrays IDOC and IRET are updated with the total number of retrievals performed (LASRET), after which a call to subroutine VRTIO updates the row header for the retrieval MD file, as well as the VASTEXT file. If no image(s) were created, this last step concludes VTPZ. However, assuming that at least one image was generated and stored in a digital area, DO LOOP 1265, which closes the digital area(s), is executed, after which VTPZ concludes by printing the message "ALL DONE" on the CRT. This concludes the discussion of the May 3, 1984 version of VTPZ.

Several important changes have been made in VTPZ since the May 3 version described in this manual. Therefore, given the importance of this program, an appendix has been included on the following 2 pages, the details of which describe the major
changes in VTPZ up to and including July 27, 1984.
VTPZ: Appendix I

This appendix describes the main changes made to VTPZ between the version discussed in the manual (May 3, 1984) and the version valid July 27, 1984.

1.) A cloud flag has been added (ICLOUD). A non-zero value (cloud contamination) precipitates a test involving band 12 and 8 TBBs, in which a failure condemns the retrieval to use only for image-generating purposes (total precipitable water vapor and (if desired) total totals index).

2.) A section of code which deals simultaneously with low level stratus cloud and temperature inversion situations has been inserted before the call to subroutine QVTWR. This section of code finds cloud pressure (MS), determines cloud top T and W values, interpolates between cloud top and level IS for the guess T and W profiles and, lastly, tests for low level stratus and temperature inversion situations.

3.) Even if a T/W retrieval fails for a given retrieval box, the VAS data for the box can still be used to create total precipitable water vapor and (if desired) total-totals image(s) (unlike the May 3, 1984 version of VTPZ).

4.) Variable GAMRET is NOT changed between the first and second retrievals (between first and second call to subroutine QVTWR) if the cloud flag (ICLOUD) has been set to 1 in subroutine FILTER.

5.) If no total precipitable water vapor value and (if desired) total-totals index are calculated for a given FOV due to an earlier retrieval failure or severe cloud contamination, a
value dependent on the band 8 TBB for that FOV (and limited to the range 155.-255., inclusive) is inserted into the image at that FOV location (The May 3, 1984 version of VTPZ simply let all the FOV's in failed retrieval boxes assume the image value 255 (white) for both total precipitable water vapor and total-totals index.
DIAGRAM 1

TYPICAL RETRIEVAL AREAS

small detector

SATELLITE COORDINATE ELEMENTS (INCREASING)

↑

SATELLITE COORDINATE LINES (INCREASING)

↑

RETRIEVAL LINE SPACING) INCIL

↑

INCIL

↑

1 FOV (288X8 KM FOR SMALL DETECTOR DATA)

4 SCAN LINES OF DATA

4 SCAN LINES OF NO S.D. DATA

3 SCAN LINES OF DATA

RETRIEVAL BOX SPACING)

NEXT RETRIEVAL LINE

NOTES: 1) DIMENSION OF RETRIEVAL BOX = 11 X 11 FOVS
2) THIS RETRIEVAL AREA HAS 4 RETRIEVAL BOXES PER RETRIEVAL LINE, WITH EACH RETRIEVAL BOX CONTAINING 121 FOVS. FOR THE FIRST RETRIEVAL LINE, THERE ARE 77 FOVS WHICH HAVE SMALL DETECTOR DATA (4 X 11 + 3 X 11), MEANING 44 FOVS HAVE NO SMALL DETECTOR DATA. FOR THE SECOND RETRIEVAL LINE, THE NUMBERS CHANGE TO 55 AND 66, RESPECTIVELY. IT IS POSSIBLE TO HAVE ANYWHERE FROM 4 TO 9 SCAN LINES OF S.D. DATA WITHIN A GIVEN 11 X 11 RETRIEVAL BOX.

large detector

↑

ELEMENTS

↑

LINES

↑

INCIL

↑

INCIL

↑

NEXT RETRIEVAL LINE

1 FOV (2 X 16 X 16 KM FOR LARGE DETECTOR DATA)

1 SCAN LINE OF DATA

NOTES: 1) DIMENSION OF RETRIEVAL BOX = 5 X 5 FOVS
2) THIS RETRIEVAL AREA HAS 4 RETRIEVAL BOXES PER RETRIEVAL LINE, WITH EACH RETRIEVAL BOX CONTAINING 25 FOVS, ALL OF WHICH CONTAIN S.D. DATA.
VTPZ

COMMON/DANGLE/VLATS, VLONS

COMMON GUESS/TGES(19), DGES(6)

COMMON LAST/LAS/LASEL, LELE, ICHAR

COMMON NAV/ VLAT, VLP, VZEN, ZSEN, IL, IE, IRAS, IPC, ITIME, JTIME, JDAY

DATA CBLNK /', 'CLET /', 'CLETG /'

DATA VMINS/999999.

DATA MAJARA/9999.

CALL CALDAY (IVER, IVI, IYM, IVO, IVMO)

CALL EMKBE ('(132),
TI, BEGIN + VTPZ, VERSION OF TI, I, M, II, AT INT', II), LAUF, IVO, IVMO, IVY, MCMN)

READ IN AREA NUMBER TO BE USED FOR STORAGE OF PRECIPITABLE WATER VAPOR IMAGE

NARA = I KWP ('MM', 1, 0)

CALL EDEST ('INVALID AREA NUMBER, MM)

CALL ABORT()

END

READ IN SECOND AREA NUMBER, TO BE USED FOR STORING IMAGE OF TOTAL-TOTALS INDEX (NOTE THAT A TOTAL-TOTALS IMAGE CANNOT BE CREATED INDEPENDENTLY; IT CAN BE GENERATED ONLY IN CONJUNCTION WITH A SIMULTANEOUS GENERATION OF A T-P-M IMAGE)

NARA = I KWP ('ARP', 1, 0)

CALL EDEST ('INVALID AREA NUMBER, MM')

CALL ABORT()

END

READ 'VASTEXT' DOCUMENTATION RECORD AND EXTRACT NEEDED INFORMATION

OPEN, READ AND CLOSE VASTEXT FILE

GET RETRIEVAL GUESS MD FILE NUMBER, AS WELL AS RETRIEVAL MD FILE AND ROW NUMBERS, FROM VASTEXT FILE

MDNG = I DOC(39)

MDRA = I DOC(40)

MDAR = I DOC(41)

CHECK FOR 'NO-OUTPUT-MEDIUM'

CALL EDEST ('NO OUTPUT MEDIUM (AREA OR MD) SPECIFIED ', 0)

CALL EDEST ('NO OUTPUT MEDIUM (AREA OR MD) SPECIFIED ', 0)
If no image of total precipitable water vapor is to be made, skip to statement 290.

Create "arareas" digital areas (files) for storage of total precipitable water vapor and (if desired) total totals index images.

Do loop 120: main loop, increments when all retrievals on a given ret. line have been completed. (il = line position of upper left corner of each ret. box; incl = line spacing between successive retrieval lines (see schematic to right)).

Do loop 1180: increments when a given retrieval on a given line has been completed (see schematic).
INITIALIZE ARRAY "TDAT" (HOLDS BRIGHTNESS TEMP. (TBD) DATA FOR ALL BANDS FOR EACH FIELD OF VIEW (FOV) IN A GIVEN RETRIEVAL BOX) TO THE VALUE "VMISO" (999999).

ALSO INITIALIZE ARRAY "IAARRAY" (HOLDS TOTAL PRECIPITABLE WATER VAPOR AND ALSO, POSSIBLY, TOTAL - TOTAL INDEX VALUES FOR EACH FIELD OF VIEW (FOV) IN GIVEN RETRIEVAL BOX TO THE VALUE "MAXNV" (255)... IT SHOULD BE MENTIONED THAT A TOTAL - TOTAL INDEX IMAGE CANNOT BE GENERATED BY ITSELF!

GET VAS TBB AND OTHER MISCELLANEOUS DATA FOR GIVEN FOV

CHECK REPORT FOR GIVEN FIELD OF VIEW TO MAKE SURE THAT;
1) ITS LATITUDE IS ≤ 90°

2) ITS SATELLITE ZENITH ANGLE (ANGLE BETWEEN L TO EARTH AT THAT POINT AND THE SAT.) IS ≤ 60°

3) IT HAS A REASONABLE SURFACE ELEVATION

4) VAS TBB DATA EXISTS FOR THAT FOV

SAVE REASONABLE TBB DATA FOR EACH BAND IN GIVEN FOV IN ARRAY TDAT

IF ANY OF THESE TESTS ARE TRUE, MOVE TO THE NEXT FOV IN THE RETRIEVAL BOX

TABULATE NUMBER OF FOVS WITH GOOD DATA IN A GIVEN RETRIEVAL BOX (MSAM) AND ALSO SUM UP SURFACE ELEVATION FOR EACH FOV IN THE BOX (SELEV)... THESE TWO VALUES USED LATER TO CALCULATE AVERAGE RETRIEVAL BOX SURFACE ELEVATION

IF NO FOVS OF DATA FOUND, JUMP TO NEXT RETRIEVAL BOX

136
ALL VAS DATA FOR GIVEN RETRIEVAL BOX ACQUIRED; NOW GET THE FIRST GUESS TO BE USED FOR THE UPWARD TEMPERATURE AND MOISTURE RETRIEVAL

CALL GESARO (IGES, MSFC, MW)

TGES(A) ≤ φ → C

CALL WMIX (FSTM, TSTM, DD, WMIX, 1)

FIND "IS" FIRST LEVEL BELOW SURFACE (PRESSURE)

AFTER THE FIRST LEVEL BELOW THE SURFACE HAS BEEN DETERMINED, VM2 COMPLETES THE GUESS T/W PROFILE PREPARATION, ASSISTING BOTH VARIABLES DOWN TO 1000 MB TO TAKE INTO ACCOUNT THE SURFACE VALUES OF T AND W(MIXING RATIO), MAKING SURE THE GUESS MIXING RATIO VALUES FALL IN THE RANGE .62 ≤ W(I) ≤ SATURATION VALUE

PRINT GUESS INFORMATION

CALL ENKODE ("T5X", "PRESSURE", * (5F7.1) ), LBUF, T5X
CALL ENKODE ("T5X", "GUESS TEMP", * (5F7.1) ), LBUF, T5X
CALL ENKODE ("T5X", "GUESS W(MIX)", * (5F7.1) ), LBUF, WMIX
CALL ENKODE ("T5X", "TSFC", "T5SFC", * (5F7.1) ), LBUF, T5SFC, T5SFC

GENERATE AVERAGE TB8S FOR THE RETRIEVAL BOX IN EACH BAND (RETURNED IN ARRAY "VOAT")

CALL FILTER (VOAT, NSPIN, ERN, MSAM, TSTM, LDETR)

MSAM = φ → C

ICOUNT = φ

I = 1 → G
I = I + 1

VOAT(A) = WMIX

STORE NUMBER OF AVERAGE TB8S CALCULATED FOR RETRIEVAL BOX FOR BANDS 1-17
DETERMINE SRC. SKIN TEMPERATURE

\[ T_{SKIN} = VSINT(VDAT, \phi, 1, TACT) \]

IF SKIN TEMP CANNOT BE CALCULATED, JUMP TO NEXT RETRIEVAL BOX AND ATTEMPT A RETRIEVAL THERE.

\[ CALL ENCODE('"VSINT \n ACTION="1", \phi, VDAT, \beta, 1, LBUG, TACT') \]

FILL ARRAY "TGS" WITH VALUES AT SELECTED LEVELS OF THE GUESS T PROFILE; CALCULATE TD AT SELECTED LEVELS FROM GUESS T AND W (MIXING RATIO) PROFILE VALUES, AND FILL ARRAY "DGS" WITH THOSE VALUES.

PERFORM "ONE STEP" MATRIX INVERSION RETRIEVAL OF T, W PROFILES.

\[ CALL GQTRA(VDAT, ERM, TOTO, TSKIN, TSTA, WSTA, IS, IAT, IFAIL) \]

IF RETRIEVAL FAILS, JUMP TO NEXT RET BOX AND ATTEMPT A RETRIEVAL THERE.

\[ \text{IFAIL} \neq \emptyset \rightarrow C \]

PERFORM RETRIEVAL AGAIN, THEREBY "FIVE-TWICE" THE RESULT FROM THE FIRST RETRIEVAL.

\[ CALL GQTRA(VDAT, ERM, TOTO, TSKIN, TSTA, WSTA, IS, IAT, IFAIL) \]

IF RET. FAILS, JUMP TO NEXT RET. BOX.

\[ \text{IFAIL} \neq \emptyset \rightarrow C \]

IF THERE WERE LESS THAN 5 NON-CLOUD-CONTAMINATED FOVS, JUMP TO STATEMENT 49.$

CALCULATE TOTAL PRECIPITABLE WATER VAPOR DOWN TO LEVEL IS.

\[ CALL PREC2(W, U, U, IS) \]

CALL TD FOR IS LEVELS BETWEEN AND INCLUDING 34KB - 1800 MB FROM RETRIEVED PROFILES OF T AND W.

\[ \text{CALL MTV}(Z, IS) \]

SET UP FOR STABILITY CALCULATION.

\[ \text{IF} \ \text{LBUG} \neq \emptyset \rightarrow \text{CALL ENCODE('"STAB. PRESS\n \ *OF IS", \ \ LBUG, ERT, U) (Z, IS, LBUG, ERT, U))} \]
Subroutines used by VTPZ:

<table>
<thead>
<tr>
<th></th>
<th>Subroutine</th>
<th></th>
<th>Subroutine</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CALDAY-I</td>
<td>53</td>
<td>READD</td>
<td>105</td>
</tr>
<tr>
<td>2</td>
<td>ENKODE-I</td>
<td>54</td>
<td>EDEST</td>
<td>106</td>
</tr>
<tr>
<td>3</td>
<td>VRTIO-I</td>
<td>55</td>
<td>MOEW</td>
<td>107</td>
</tr>
<tr>
<td>4</td>
<td>TQSET-I</td>
<td>56</td>
<td>CLEANW</td>
<td>108</td>
</tr>
<tr>
<td>5</td>
<td>GETDAY-I</td>
<td>57</td>
<td>TQ</td>
<td>109</td>
</tr>
<tr>
<td>6</td>
<td>GETFRM-I</td>
<td>58</td>
<td>OUTINT</td>
<td>110</td>
</tr>
<tr>
<td>7</td>
<td>TVSAT-I</td>
<td>59</td>
<td>II</td>
<td>111</td>
</tr>
<tr>
<td>8</td>
<td>VASDAT-I</td>
<td>60</td>
<td>SATEAR</td>
<td>112</td>
</tr>
<tr>
<td>9</td>
<td>ARASIZ-I</td>
<td>61</td>
<td>SATPOS</td>
<td>113</td>
</tr>
<tr>
<td>10</td>
<td>ENAREA-I</td>
<td>62</td>
<td>ANGGET</td>
<td>114</td>
</tr>
<tr>
<td>11</td>
<td>OPNA-I</td>
<td>63</td>
<td>SOLARP</td>
<td>115</td>
</tr>
<tr>
<td>12</td>
<td>GESPRO-I</td>
<td>64</td>
<td>ANGLES</td>
<td>116</td>
</tr>
<tr>
<td>13</td>
<td>WMIX-I</td>
<td>65</td>
<td>SATPOS</td>
<td>117</td>
</tr>
<tr>
<td>14</td>
<td>FILTER-I</td>
<td>66</td>
<td>VROPO</td>
<td>118</td>
</tr>
<tr>
<td>15</td>
<td>QTTRW-I</td>
<td>67</td>
<td>NVINIT</td>
<td>119</td>
</tr>
<tr>
<td>16</td>
<td>PRECW-I</td>
<td>68</td>
<td>GETNAV</td>
<td>120</td>
</tr>
<tr>
<td>17</td>
<td>HTV-I</td>
<td>69</td>
<td>EPOC</td>
<td>121</td>
</tr>
<tr>
<td>18</td>
<td>SNDANL-I</td>
<td>70</td>
<td>GETGAM</td>
<td>122</td>
</tr>
<tr>
<td>19</td>
<td>PACK-I</td>
<td>71</td>
<td>VASEAV</td>
<td>123</td>
</tr>
<tr>
<td>20</td>
<td>WRITA-I</td>
<td>72</td>
<td>DDST</td>
<td>124</td>
</tr>
<tr>
<td>21</td>
<td>EDEST-I</td>
<td>73</td>
<td>PLNKIV</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>CLOSE-1</td>
<td>74</td>
<td>LOPEN</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>ABORT-I</td>
<td>75</td>
<td>MOVC</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>ENCODX</td>
<td>76</td>
<td>LWGET</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>CONTNT</td>
<td>77</td>
<td>READOF</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>LTQ</td>
<td>78</td>
<td>RBYTSX</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>MOV8</td>
<td>79</td>
<td>ZEROS</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>CLEANA</td>
<td>80</td>
<td>RDTRK</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>TQ</td>
<td>81</td>
<td>MOVC</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>PRLINKX</td>
<td>82</td>
<td>READDL</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>PRCLOS</td>
<td>83</td>
<td>LGCLOS</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>LOCK</td>
<td>84</td>
<td>LWMP</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>PRMR</td>
<td>85</td>
<td>LWPO</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>PRRD</td>
<td>86</td>
<td>LWSO</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>UNLOCK</td>
<td>87</td>
<td>JMBWF</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>PRCL</td>
<td>88</td>
<td>LWNEWF</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>POST</td>
<td>89</td>
<td>GOTTIM</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>PROPEN</td>
<td>90</td>
<td>WRITDU</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>BLKA</td>
<td>91</td>
<td>RAOBIN</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>PRPRP</td>
<td>92</td>
<td>CSHAOB</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>ZECNV</td>
<td>93</td>
<td>CSHAOR</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>DOPEN</td>
<td>94</td>
<td>CSHAOB</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>DREAD</td>
<td>95</td>
<td>CSHAOS</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>ENCODE</td>
<td>96</td>
<td>CSHAOM</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>MOVWC</td>
<td>97</td>
<td>CSHAOS</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>DWRIE</td>
<td>98</td>
<td>CSHAOP</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>DCLOSE</td>
<td>99</td>
<td>CSHAORI</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>TRMNRL</td>
<td>100</td>
<td>MNRAOB</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>TQCMES</td>
<td>101</td>
<td>INTPTW</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>TOC</td>
<td>102</td>
<td>EXTEMP</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>STC</td>
<td>103</td>
<td>STDATM</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>SDEST</td>
<td>104</td>
<td>CLMGES</td>
<td></td>
</tr>
</tbody>
</table>
PLVA

PLVA is used to plot either VAS retrieval or surface MD file data on the video screen over the satellite image. Several different types of data can be displayed. For instance, the user can display a certain parameter at a given level, differences between retrieval and first guess for either temperature or dew point at a given level, non-level parameters such as brightness temperature or relative humidity, layer thickness, or winds.

Initially, data such as debug option, graphics and image frame numbers (IWRM and IFRM, respectively), the VASTEXT file, etc. are read in. Then, after variables IRAS and IPIC have been determined, the satellite coordinates (line, element) of the cursor are set via subroutine TVSAT, the navigation is initialized, and the coordinates of the image/sounder area (LLNW, LLSE) are filled from the VASTEXT file via IDOC (25) and (26), respectively. These coordinates are then used to determine the N, S, W and E boundaries for the data to be plotted. Note that these latitude/longitude boundaries can also be included in the program keyin via keyword parameters LAT and LON. Now, the MD file and row from which the user wishes data to be taken and plotted is set (MDNO, MDR), with the default values being the retrieval MD file and row numbers taken from the VASTEXT file (IDOC (40) and (41), respectively).

Following this, the MD file is opened for read/write, and the row header for the file is read. The data in the row header includes, among other things, the number of reports which exist in that particular row. Then, a test is conducted (READ TEST
RECORD) to make sure data actually exists in this row. Note also that the maximum possible number of data entries in a given MD file row is stored in variable MMAX (MMAX=MDHD(5)). (The array MDHD was filled via a previous call to function MDINFO). Following this, the keys for the MD file are read, which becomes vital when the actual data is accessed from the file later on.

Next, after the size of the data to be plotted is determined and the plot package is initialized, the first and second positional parameters are read in via the three statements CCHR=CPP(1,'Z '), ILEV=IPP(2,0) * 10 and CLEV=CPP(2,' '). Note the different values which variable ISUB assumes, depending on which data quantity is being plotted. The information loaded into these three variables will determine both WHAT type of data is to be plotted (Z, T, brightness temperature, etc.), as well as HOW the data should be plotted (either as values or differences). The default values (if neither of the first two positional parameters are keyed in) are Z and 500, respectively, meaning a 500 mb height field will be plotted on the video screen if PLVA is keyed in by itself. In addition, variable IL2 stores the second (lower) level in case a thickness plot is desired. Then, the data accessed previously by function MDKEYS is used in DO LOOP 18 to determine which words in any given MD file report will contain such quantities as latitude, longitude, etc. Next, the number of reports to be read (not necessarily plotted) from row MDR of the MD file is determined and placed in variable MREC.

Up to this point, PLVA has functioned quite similarly for all the possible plotting options. However, as PLVA now enters
an implicit DO LOOP (which runs approximately from code line 185 (15 CONTINUE) to 320 (IF(MM.LE.MMAX) GO TO 15)), its operation varies, depending on which parameter is being plotted. Therefore, the discussion will henceforth deal separately with different individual options.

Initially, after the first report has been read (I0K=MDGET(MDNO, MDR, MM, IOUT)), there is an option to skip a given report if it has been previously rejected, or plot all reports, regardless of previous rejection (IF(IOUT(NMOD).NE. 0.AND.IALL.EQ.0) GO TO 70). These two steps are done for EVERY option. Then, if the pressure level (ILEV) has been set to some non-zero integer, and the plot is to consist of a non-difference variable (T, Z, etc.), the words of the array IOUT storing the correct level and character to be plotted are calculated in DO LOOPS 45 and 47. Following this, assuming the retrieval report is within the previously-defined plotting area, the report line/element (FLIN,FELE) and then raster/pixel (IRAS,IPIC) coordinates are determined in succession by subroutines SATEAR and SATTV. Finally, the color in which the data will be plotted is established, and the actual data plot itself (note that ISUB=0, and that the value plotted is stored in variable IDAT) is carried out. The subroutine which does the actual plotting is PLTDIG.

As another case, if the user is plotting relative humidity or brightness temperature values (no level keyed in), the word of the array IOUT which stores the variable to be plotted is again determined in DO LOOP 43, after which the same procedure as above
is once again followed.

If thicknesses are to be plotted, the words within IOUT containing the first (upper) pressure level and character (Z) are determined in DO LOOPS 45 and 47, with the same process being repeated for the second (lower) pressure level and character in DO LOOPS 52 and 54. (Note that variable IL2 will have been given a non-zero value (IL2=IKWP('LEV2',1,0)*10) just before DO LOOP 18 was entered.) Following this, assuming the data is not missing for either pressure level and that the retrieval is within the latitude/longitude boundaries, the retrieval raster/pixel coordinates are determined via SATEAR and SATTV, and the thickness to be plotted is calculated (IF(IAD2.NE.0) IDAT=IDAT-IOUT(IAD2)/ISC). Following this, the same steps as above are followed.

On the other hand, if difference plots between retrieval and guess for temperature or dewpoint are desired, DO LOOPS 45 and 47 are again executed, and the raster/pixel coordinates and plotting color are again calculated. Then, the difference quantity is formed immediately beyond statement 60 (IDAT=IDAT-IOUT(IADD+ISUB)/ISC). (The user should note at this point, in regard to DO LOOP 47 and the difference quantity calculation, that variable CCHR will have been reset to T or TD, depending on whether TDIF or DDIF was keyed in, and that variable ISUB will have been given a value of 5, back in the vicinity of statement 20.) Subsequent to the difference quantity calculation, the difference quantity itself is plotted on the video screen.

Finally, PLVA can also be used to plot surface data, such as
winds or other variables. The user should key in only the appropriate character positional parameter (no level positional parameter keyed in) after PLVA (PLVA (space) character) to force the program to operate in this mode. For example, when the user desires a plot of surface winds (PLVA WIN), PLVA executes DO LOOP 43, jumps to statement 56, and calculates TV coordinates, color and the data value to be plotted (IDAT). Then, since variable LCHR .EQ. WIN, and variable ISS .NE. 0 (ISS = satellite identification number, and is .NE. 0 for GOES satellites), subroutine DIRADJ is called. This subroutine adjusts the wind direction so it is from its true direction when plotted on the satellite image (Note that true north, for example, is different on the video screen at different image longitude locations along a given image latitude.) Finally, subroutine BARB does the actual plotting.

Essentially, the main point to remember about PLVA is that the user has to worry only about the parameter itself (not the level) when plotting surface data or non-level data (such as relative humidity or brightness temperature), while he/she must worry about both parameter AND level when plotting Z, T, TD, retrieval-guess differences for T or TD, etc. at a given level other than the surface. The only exception to this rule pertains to a surface data plot of Z (height), where the user must key in SFC as the second positional parameter (see code detailing default case when neither first nor second positional parameters have been keyed in).

At this point, a few sample keyins will be presented.
Hopefully, these commands will enable the user to become more familiar with the operation of PLVA.

1.) PLVA Z 500: Plot 500 mb heights, taking data from retrieval MD file and row as listed in VASTEXT file.

2.) PLVA RH3 X MDNR=2615 MDRR=1: Plot upper level (band 10) relative humidity values, taking data from row 1 of retrieval MD file 2615.

3.) PLVA TDIFF 700: Plot difference in temperature between retrieval and guess at 700 mb, taking data from retrieval MD file and row as listed in VASTEXT file.

4.) PLVA V06: Plot retrieval box-averaged band 6 brightness temperatures (measured, not retrieved) from the VASTEXT file-defined row of the retrieval MD file.

5.) PLVA Z 300 LEV2=500: Plot 300-500 mb thicknesses, taking data again from the VASTEXT file-defined retrieval MD file.

6.) PLVA TSL MDNR=4183 MDRR=12: Plot mean sea level temperature from specified retrieval surface MD file and row (plotted data will be from 12 GMT).
DO LOOP IS DETERMINES WHICH WORDS OF MD FILE REPORT ARRAY "IOUT" CONTAIN QUANTITIES SUCH AS LATITUDE, LONGITUDE, NUMBER OF REPORTS IN THAT ROW OF THE MD FILE, MOD (IGNORE REPORT) FLAG, RETRIEVAL TYPE (NTYP), ETC. STORE NUMBER OF REPORTS IN ROW "MOD" OF MD FILE IN VARIABLE "HREC"
DETERMINE COLOR DATA WILL BE PLOTTED IN THICKNESS CALCULATION

PREPARE FOR WIND DATA PLOT

ADJUST WIND DIRECTION SO FORTHCOMING WIND BARB PLOT WILL BE FROM TRUE DIRECTION ANYWHERE ON IMAGE CALL OIRADJ(FLAT, FLO, SATON, DIR, ADJ)

CALL BARB(ADJ, SPD, XP, YP, KOLOR, SZ)

ISUB = φ: DIFFERENCE QUANTITY IS TO BE PLOTTED, AND VICE-VERSA

FORM DIFFERENCE BETWEEN RETRIEVAL OR GUESS FOR TEMPERATURE OR DEWPOINT

FORM DIFFERENCE QUANTITY

PLOT DATA CALL PLTDIG(IRAS + MAG, IPIC + MAG, IDAT, MAG, 1, KOLOR)

M = M + 1

M > MREC

MM = MM + 1

RETURN TO PROCESS NEXT REPORT

CALL ENDPDLT END

above to convert to centigrade
### Subroutines used by PLVA:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DOPEN-I</td>
<td>53</td>
</tr>
<tr>
<td>2</td>
<td>DREAD-I</td>
<td>54</td>
</tr>
<tr>
<td>3</td>
<td>DCLOSE-I</td>
<td>55</td>
</tr>
<tr>
<td>4</td>
<td>TVSAT-I</td>
<td>56</td>
</tr>
<tr>
<td>5</td>
<td>NVINIT-I</td>
<td>57</td>
</tr>
<tr>
<td>6</td>
<td>TSNIO-I</td>
<td>58</td>
</tr>
<tr>
<td>7</td>
<td>OUTINT-I</td>
<td>59</td>
</tr>
<tr>
<td>8</td>
<td>GETFRM-I</td>
<td>60</td>
</tr>
<tr>
<td>9</td>
<td>INITPL-I</td>
<td>61</td>
</tr>
<tr>
<td>10</td>
<td>SATEAR-I</td>
<td>62</td>
</tr>
<tr>
<td>11</td>
<td>SRCH-I</td>
<td>63</td>
</tr>
<tr>
<td>12</td>
<td>SATTV-I</td>
<td>64</td>
</tr>
<tr>
<td>13</td>
<td>PLTDIG-I</td>
<td>65</td>
</tr>
<tr>
<td>14</td>
<td>ENDPRT-I</td>
<td>66</td>
</tr>
<tr>
<td>15</td>
<td>SDEST-I</td>
<td>67</td>
</tr>
<tr>
<td>16</td>
<td>DIRADJ-I</td>
<td>68</td>
</tr>
<tr>
<td>17</td>
<td>BARG-I</td>
<td>69</td>
</tr>
<tr>
<td>18</td>
<td>ENCODE</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>ENCODX</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>ABORT</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>CONTNT</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>LTQ</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>ZECONV</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>MOV</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>CLEANA</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>TQ</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>PRLINX</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>PRLOS</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>PROPEN</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>PRPRPR</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>LOCK</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>PRWR</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>PRRD</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>UNLOCK</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>PRCL</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>POST</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>BLKA</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>TRMNL</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>TQMS</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>STC</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>ITOC</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>GETNAV</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>EPOCH</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>GETGAM</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>VASNAV</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>DDEST</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>MOVCH</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>CLEANEW</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>EDEST</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>MOV</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>II</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>DWRITE</td>
<td></td>
</tr>
</tbody>
</table>
The purpose of BNVA is to create grids of different meteorological parameters. These parameters can be either from retrieval MD or other types of MD files. Some of the parameters that can be gridded with BNVA include differences between retrieval and guess for temperature (T) and dewpoint (TD) (brightness temperature (TBB) differences between retrieval and guess cannot be gridded as of this time, due to the structure of retrieval MD file reports). In addition, one can also generate grids of T, TD, TBB, relative humidity, thickness, winds or other quantities which exist in the MD file from which the data is being taken. Before the user runs BNVA, the gridfile he/she wishes the results to be stored in must be set with system program IGU to make sure the final grid is stored in the desired gridfile. (The reason for this will become more clear as the discussion proceeds.)

There are two first guess options which can be used in BNVA, in addition to the option of not using ANY first guess. One of these is the grid first guess, while the other is the guess MD file first guess. A guess is advisable if the data to be gridded is sporadic in coverage over the area of the desired grid. In such situations, the generation of a grid without using a first guess could lead to a poor analysis. In addition, note that each guess consists only of heights, temperatures or dewpoints, depending upon which parameter is being gridded, so the user must keep in mind that a first guess cannot be used when generating grids of TBBs, thicknesses, winds, differences between retrieval
and guess for T or TD, etc.

Initially, things such as the final output grid number (NGOUT) and the debug option (KBUG) are set, after which the VASTEXT file is opened, read and closed. At this point, several other variables are determined, such as the retrieval MD file and row numbers (MDNO and MDR, respectively), the upper air guess grid file number (NGFG), and the upper air guess MD file number (MDNG). All four of these variables have VASTEXT defaults if no keyword parameter value has been designated. Now, after the parameter to be gridded (ICHR) and the level at which it is to be gridded (ILEV) have been determined, the MD file (if it exists -- checked via function MDINFO) is opened for read/write, the row header is read, a test to make sure that data exists in the MD file is performed ("READ TEST RECORD"), and a call to function MDKEYS reads the keys contained within the schema for that particular MD file. At this point, the elements within the MD file report array for the number of retrievals or reports in a given row of the MD file (variable NCA), latitude, longitude, and mod flag (variable NMOD -- MD file report array element NMOD will indicate whether a given retrieval or report is good) are determined in DO LOOP 2. The variable which will store the number of reports, NRPT, is then filled following DO LOOP 2.

The following lines of code set several variables, including variable ISUB, which equals 5 for T or TD differences between retrieval and guess. Then, keyword parameter LEV2 is evaluated to see if thickness calculations and a resultant grid generation thereof is desired, and variable NGG is filled with the number of
the guess grid to be used in the event a guess grid option is desired. Note that variable CLEV is unimportant at this time (5/1/84), since TBB differences between retrieval and guess cannot be done, and surface grids of retrieval minus guess differences for temperature or dewpoint do not require a second keyed-in positional parameter.

From this point forward (IF (NGG.NE.0) GO TO 14), the discussion of BNVA will be broken down into separate discussions for each of the three program options: guess grid, guess MD file or no guess.

First, in the case when the guess grid option is being used, BNVA moves to statement 14 (IF (NGG.NE.0) GO TO 14). Then, after the analysis field (the array IDL, in this case) has been initialized to zero in DO LOOP 12, the guess grid is accessed from the guess grid file (NGFG) by function IGET. The user must make sure the guess grid being accessed is no larger than 2400 words; otherwise, IOK will be less than 0, and BNVA will terminate. Following this, the number of gridpoints (NPTS) is calculated, the guess grid data is transferred to the array FLD, the latitude/longitude dimensions of the grid are determined (LAN/TLAT, LAS/SLAT, LOW/WLON and LOE/ELON), and information pertaining to number of rows, number of columns and grid increment is stored in KOUT(2), (3) and (4), respectively. In addition, if KBUG .GE. 0, a call to subroutine OUTINT will display this information on the CRT. Then, after a jump to statement 110 (remember, this is a discussion of the guess grid option), and if keyword parameter NODAT was keyed in .NE. 0, BNVA
will jump to statement 150, and the guess grid just accessed will be stored in gridfile NGFG by function IGP3UT, after which BNVA will terminate.

Normally, however, NODAT = 0, and the program enters an implicit DO LOOP ("BEGIN IMPLICIT DO LOOP ON REPORTS"), the broad purpose of which is to read data from MD file MDNO and prepare the input data for the subsequent grid generation. The length of this loop is approximately 130 lines of code.

Initially, after a given MD file report is read (I0K=MDGET(MDNO,MDR,NN,KBUF)), and assuming it has not been previously earmarked for rejection by the mod flag (KBUF(NMOD).EQ. 0), DO LOOPS 43, 45, 47, 52 and 54 determine (in terms of level and parameter) which data will be taken from each MD file report and used in the subsequent grid generation. Following these loops, and assuming the raw MD file data is not missing (KBUF(IADD/IAD2) is .NE. MISS), the data to be gridded (nearly in final form) is stored in variable IDAT (IDAT=KBUF(IADD)). The final data value to be gridded will then be calculated and stored in variable DAT.

If the correct data cannot be accessed (in terms of getting an MD file report during the original MDGET, or in terms of matching the pressure level/character during the DO LOOP sequence), the implicit loop will jump to statement 148, increment variable NN, and jump back to statement 15 to access the next MD file data report, meaning THAT particular report will not be used when generating the grid. Other conditions causing a jump to statement 148 include missing data (IF
(KBUF(IADD).EQ.MISS) GO TO 148), or location of the MD file
report outside the grid boundaries (IF FLAT.GT.
TLAT.OR.FLAT.LT.SLAT) GO TO 148).

However, assuming the raw MD file data is gathered and the
value to be gridded is determined with no problems, the final
data to be gridded, and the latitude/longitude coordinates of the
data, are stored in the arrays DA, STLAT and STLON, respectively.
Now, since IGB is .NE. 0, and assuming that keyword parameter
EDIT was not keyed in (IKWP('EDIT',1,0).EQ.0), a gross error
check is performed which compares the value to be gridded (DAT)
with a value taken from the guess grid (VAL).

If a given MD file report fails the gross error test, BNVA
jumps to statement 148, again meaning THAT particular MD file
report will not be used in the Barnes analysis. However, assuming
the test passes, the program jumps to statement 70, where the
Barnes arrays DAS, RW and CL are filled. Then, after the
variables SUM and SUMS have been updated, and assuming KK .LE.
NRPT and that there are less than 5000 data values generated so
far, variable NN is incremented and, unless NN .GT. MMAX, BNVA
jumps back to statement 15 to look for the next report in row MDR
of the MD file.

After all the MD data has been read and prepared for the
Barnes analysis, BNVA jumps to statement 140 (usually via the
statement IF (KK.GT.NRPT) GO TO 140), where a standard deviation
(ISD) is determined both from the data to be gridded and
previously-set ERR values (see code immediately preceding the
implicit DO LOOP). Then, a Barnes analysis (subroutine FBARN) is
executed, which produces a uniform set of gridpoint values based on the data to be gridded and the guess grid. The Barnes analysis makes only 1 pass if the guess grid or guess MD file option is being used, while 2 passes are made for the no-guess option. The final data from the Barnes analysis is then stored in the array IDL in DO LOOP 155.

Following this, if the edit option is NOT in operation (IKWP ('EDIT',1,0).EQ. 0), the grid is written into gridfile NGFG in the next empty slot after NGOUT (see description of function IGPUS in appendxi IV). At this point, if the pass weights are not to be written into the same gridfile, BNVA is finished. However, if the weights for the one pass of the Barnes analysis are to be written into the gridfile (IKWP('WGT',1,0).NE. 0), one more call to function IGPUS accomplishes this task, after which BNVA terminates.

On the other hand, if the edit option IS in operation (IKWP('EDIT',1,0).GT.0), a value determined from the final grid (VAL) is compared to the original data value from before the Barnes analysis (DAT), and if the difference is .GE. variable ERR, the element of the array KBUF which indicates rejection (NMOD...the mod flag) is set to a non-zero value. Then, the MD file is updated via function MDPUT to reflect this change. Note that the Barnes analysis grid will not be written to an output gridfile if the edit option is in effect (see the RETURN immediately following DO LOOP 130). In other words, the only purpose of the edit option is to edit the source MD file data.

Returning to the statement IF (NGG.NE.0) GO TO 14, I will
now discuss briefly the guess MD file option of BNVA. First, since NGC = 0, the boundaries of the grid (LN, LS, LW and LE) are set by keyed-in values of LAT and(or) LON, or by VASTEXT default values if no keyin of LAT/LON exists. Then, the grid increment (in tenths of degrees) is determined (INC). If the grid increment has been keyed in (LINC .NE. 0), the final number of gridpoints (NPTS) is not restricted (can be .GT. 2400). However, if the grid increment has not been keyed in (LINC .EQ. 0), the total number of gridpoints must be .LE. 2400. Otherwise, the grid increment is increased by 1 (less resultant gridpoints), and control returns to statement 13.

Once the grid increment has been determined, the analysis field is initialized to 0, a jump to statement 105 is taken, variables TLAT, SLAT, WLON and ELON (bounds for using MD file data in the Barnes analysis) are set, and the guess data is read from the guess MD file (MDNG), with the surface option (NOSFC) taken into account, via subroutine ANGSS. Then, assuming keyword parameter NODAT has not been keyed in (IKWP('NODAT',1,0).EQ.0), variables ERR/FCK are set, and the implicit DO LOOP accesses data from the MD file and prepares it in a like manner to that described earlier for the guess grid option. The final data and their latitude/longitude positions are again stored in the arrays DA, STLAT and STLON, respectively. Next, since IGB is .NE. 0, and assuming keyword parameter EDIT was not keyed in, the gross error check is performed, which compares the value to be gridded (DAT) with a value taken this time from the guess MD file (VAL). If a given MD file report fails the gross error test, BNVA jumps to
statement 148. On the other hand, if the test does pass, the Barnes arrays are filled, variables SUM and SUMS are updated, and BNVA returns to statement 15 for more MD file data. After all the MD file reports have been read and the data to be gridded determined, BNVA jumps to statement 140 (KK becomes .GT. NRPT), where the standard deviation of the data to be gridded is calculated, and the Barnes analysis is performed. From this point to the end, BNVA via the guess MD file option functions identically to the guess grid option, except the destination gridfile number for the final Barnes analysis grid is given by word 6 of User Common (UC(6)... set before the execution of BNVA via system program IGU, and accessed within BNVA itself by function LUC), instead of the variable NGFC.

Returning once more to the statement IF (NGG.NE.0) GO TO 14, the no guess option of BNVA will now be described. This version should be used only if the MD file data coverage is fairly complete over the grid area.

Beginning with this statement, BNVA progresses with the no guess option precisely as the guess MD file option down through the initialization of the analysis field. Then, after jumping to statement 105 (note that NGG=0 in this case), the bounds for using MD file data in the grid generation (TLAT, SLAT, Wلون and ELON) are set, variable CGS is established (CGS=CKWP('GSS',1,'')) and, since CGS will equal '', BNVA jumps to statement 110. At this point, keyword parameter NODAT is checked and, assuming NODAT was not keyed in, the program sets variables ERR/FCK and then passes directly into the implicit DO LOOP. Within the loop,
a given MD file report is read via function MDGET, and DO LOOPS 43, 45, 47, 52 and 54 again evaluate which particular data will be taken from a given MD file report and used in the grid generation. (Note that any type of MD file data possible with BNVA (see introductory paragraph) can be gridded with the no guess option.) Then, assuming all the raw data exists (KBUF(IADD) and, if needed, (IAD2).NE. MISS), the data value to be used in the grid is calculated and stored in variable IDAT and then subsequently in variable DAT. IDAT will be calculated via IDAT=KBUF(IADD), for single-level data, IDAT=IDAT- KBUF(IAD2), for thicknesses, or IDAT=IDAT- KBUF(IADD+ISUB), for retrieval-guess differences of T or TD. At this point, the arrays DA, STLAT and STlon are again filled, and BNVA jumps to statement 146 (IF (IGB.EQ.0) GO TO 146) where, if the character to be gridded is wind, the appropriate U or V component is determined (if U or V was keyed in as the character to be gridded, ICHR='SPD', LCHR='U' or 'V'). Following this, the Barnes arrays DAS, RW and CL are filled, and variables SUM and SUMS are updated. Then, after the implicit DO LOOP has been exited, the standard deviation of the data to be gridded is calculated, and the Barnes analysis is performed by subroutine FBARN. From this point to the end, BNVA processes the final gridded data identically to the guess MD file option, with the exception that 2 pass weight arrays can be stored in gridfile NGRF with the no-guess option, as opposed to only 1 with the guess MD file (and guess grid) options, due to the fact the Barnes analysis makes 2 passes for the no-guess option.
Since there might be some confusion concerning the different gridding options possible with BNVA, I will now present a few sample keyins and the associated response of BNVA.

1.) BNVA T 700 19: grid 700 mb retrieval temperatures taken from MD file whose number is defined in the VASTEXT file, using the guess contained in grid #19 of the upper air guess gridfile defined in the VASTEXT file; results will be stored after the last grid in the same upper air guess gridfile.

2.) BNVA Z 500 GSS=G INC=10: grid 500 mb retrieval heights taken from the retrieval MD file defined in the VASTEXT file, using a first guess taken from the upper air guess MD file (whose number is also defined in the VASTEXT file); final grid will have spacing of 1 degree latitude/longitude, and will be stored in the gridfile whose number is defined in word 6 of User Common.

3.) BNVA T 500: grid 500 mb retrieval temperatures, with no first guess, with the data being taken again from the VASTEXT file-defined retrieval MD file; results will be stored in gridfile whose number is defined in word 6 of User Common.

4.) BNVA T 500 MDNR=4184: same as above, except raw data will be taken from MD file 4184.
5.) BNVA Z 500 LEV2=850: grid 850-500 mb thicknesses, using no first guess, with the data being taken from the VASTEXT file-defined retrieval MD file; results stored in gridfile whose number is defined in word 6 of User Common.

6.) BNVA T: grid surface temperature, using no first guess, with the data being taken from the MD file whose number is defined in the VASTEXT file; store results in gridfile whose number is defined via word 6 of User Common.

7.) BNVA TDIF 700: grid 700 mb retrieval minus guess differences for temperature, with data taken from VASTEXT file-defined retrieval MD file; results stored in gridfile whose number is defined in word 6 of User Common.
IF "DIFF" KEYED IN FOR 2ND POSITIONAL PARAMETER, GRID WILL BE OF TAB DIFFERENCES BETWEEN RETRIEVAL AND GUESS; IF "SRC" KEYED IN, GRID WILL BE OF PARAMETER DIFFERENCES BETWEEN RETRIEVAL AND GUESS AT THE SURFACE.

CLEV = CPP(2,0)*10

IF ICHR = 'T' AND ILEV = 0 AND CLEV = ' ' 

ILEV = 5000

LPR = ILEV/10

CHECK FOR THICKNESS CALCULATION

IL2 = IKWP(IELV2,1,0)*10

CHECK FOR T88 DIFF

CHECK FOR GUESS GRID

READ IN GUESS GRID, IN THE EVENT THE GUESS GRID OPTION IS BEING USED
SEE IF 2ND LEVEL (MUST BE LOWER LEVEL) FOR THICKNESS CALCULATIONS IS A NON-SURFACE LEVEL

TAKE CARE OF CASE WHEN 2ND LEVEL IS THE SURFACE

\[ G \]

\[ \text{IF } L2 = \text{KBUF}(x) \]

\[ F \]

\[ BL2 = \text{CLEV}_\text{LT}(\text{KBUFF}) \]

\[ F \]

\[ IAD2 = \text{LOCOS}(n) \]

DETERMINE PARAMETER TO BE GRIDDED; USED TO MAKE SURE HEIGHT DATA EXISTS FOR THE LOWER LEVEL

\[ H \]

\[ \text{IF } L2 \text{IF } \text{LIT0CM} = \text{LIST}(n) \]

\[ F \]

\[ 148 \]

\[ C \]

\[ \text{IF } \text{KBUFF} \text{IF } \text{IAD2} = \text{MISS} \]

\[ 148 \]

\[ C \]

\[ \text{IF } \text{IADD} = \text{LOCOS}(n) \]

\[ C \]

\[ \text{IF } \text{KBUFF} > 1 \]

CALL OUTINT(OUT)

\[ C \]

\[ \text{IF } \text{KBUFF} \text{IF } \text{IAD2} = \text{MISS} \]

\[ 148 \]

\[ C \]

\[ \text{FLAT} = \text{KBUFF} * \phi \]

\[ \text{FLOM} = \text{KBUFF} * \phi \]

MAKE SURE MD FILE REPORT IS WITHIN BOUNDS FOR GRID GENERATION

\[ \text{IF } \text{FLAT} > \text{FLAT} \text{OR } \text{FLAT} \text{OR } \text{FLAT} \]

\[ 148 \]

\[ C \]

\[ 173 \]
DETERMINE DIFFERENCE TO BE GRIDDED BETWEEN RETRIEVAL AND FIRST GUESS (TABS OR TO DIFFERENCES)

FILL ARRAYS DA, SLAT AND SLON

DAT = FLOAT(IDAT)/FSCL

ABOVE ARE SAVED FOR EDITING

ABOVE SAVES ORIGINAL SUBSCRIPT FOR EDITING

CALL VALUE (LAN, LAS, LON, INC, IDL, FSCL, VAL)

ABOVE IS GROSS ERROR CHECK

CALL OUTINT(KOUT)

IF FIRST POSITIONAL PARAMETER WAS EITHER "U" OR "V", CALCULATE THAT PARTICULAR WIND COMPONENT

FILL ARRAY FOR BARNES

FILL BARNES ARRAYS
EDIT OPTION WILL BE IN EFFECT IF KEYWORD PARAMETER "EDIT" WAS KEPT IN WITH A VALUE > 0.

IN LOOP 1:30: COMPARE VALUE FROM FINAL GRID TO INITIAL MD FILE VALUE, SETTING MD FLAG FOR MD FILE REPORT TO "MMD" (REPORT EFFECTIVELY EDITED FROM MD FILE) IF VALUES DIFFER BY "ERR" OR MORE.

CALL VALUE (LNV, LAN, LOW, LON, IDL, FSCL, VAL)

GROSS ERROR CHECK

IF MD FILE REPORT VALUE IS BAD, GET REPORT FROM MD FILE, SET MD FLAG, AND PUT EDITED REPORT BACK INTO MD FILE.

CALL ERRCODE(+1, 4, 2F, 2, 3F, 1), LNV, NN, FLAT, FLOW, DAT, VAL, D

END

DETERMINE GRIDFILE IN WHICH TO STORE OUTPUT GRID OF VALUES, THICKNESSES OR OFFSEE BETWEEN RETRIEVAL AND GUESS.

PUT GRID INTO OUTPUT GRIDFILE VIA FUNCTION "IGOUT"

CALL SDECST("ANALYSIS FILED AS GRID NO., ISTAT")

STOR 1ST SET OF PASS WEIGHTS FROM BARNES ANALYSIS VIA FUNCTION "IGOUT"

STOR 2ND SET OF PASS WEIGHTS FROM BARNES ANALYSIS ALSO VIA FUNCTION "IGOUT"
CALL SDEST('WEIGHT1 FILED AS GRID NO.', ISTAT)

CALL SDEST('WEIGHT2 FILED AS GRID NO.', ISTAT)

KBUG \( \neq \) \( \emptyset \)  \( \rightarrow \) F

KBUG \( \neq \) \( \emptyset \)  \( \rightarrow \) F

END
Subroutines used by BNVA:

<table>
<thead>
<tr>
<th></th>
<th>Subroutine</th>
<th></th>
<th>Subroutine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SDEST-I</td>
<td>53</td>
<td>LWSO</td>
</tr>
<tr>
<td>2</td>
<td>TSNIO-I</td>
<td>54</td>
<td>JMBWTF</td>
</tr>
<tr>
<td>3</td>
<td>DOPEN-I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>DREAD-I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DCLOSE-I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>OUTINT-I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ANGS-I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>FBARN-I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>VALUE-I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ENKODE-I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>BLKA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MOVCW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>MOVB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>LTQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>CLEANA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>TQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>PRLNX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>PRCLS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>PROPEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>PRPRFR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>LOCK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>PRWR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>PRRD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>UNLOCK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>PRCL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>POST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>ABORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>II</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>STC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>ENCODE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>ENCODX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>CONTNT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>ZECNV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>DWRITE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>LWCLS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>MOVWC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>LWMOP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>ITOC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>VAGES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>SURGES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>GETSFV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>IGNAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>WMIX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>FILL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>LINFIL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>INTER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>HEAPFY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>LWPO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>EDEST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>MOVW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>CLEANW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>LWNEWF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Program GWVA is used to produce wind vectors from the previously-calculated height grids generated by program BNVA. In addition, a plot of the resultant wind vectors is included as an option.

The first things done in GWVA are to initialize the plot routines using subroutine INITPL, set the plot option (IPLT), and open/read the VASTEXT file. Assuming the plot option is in effect (IPLT=0), subroutine TVSAT returns variables ISS and JD, which are required for the navigation initialization process, after which the navigation file is opened via subroutine NVINIT. Then, variables MDNO and MDR are set to the retrieval MD file and row numbers, respectively, as listed in the VASTEXT file. After the retrieval MD file is opened for read/write, the wind type to be calculated is determined by reading in the value associated with keyword parameter TYPE. There are four types of wind calculations possible with GWVA: GR (gradient), G (geostrophic), AG (ageostrophic), and IS (isallobaric). If no wind type is keyed in, the program will produce gradient winds by default.

In the following code, information such as color of plotted barbs (variable KOL--default = 1 (red)), grid file (NGRF), and grid number within the gridfile (NGR) are set, after which the height grid from which the winds are to be produced is accessed via function IGETG. Note that the gridfile from which the height grid is to be taken must be set before GWVA is executed. This can be done by using system program IGRU ("IGU SET 4190", for example). Further code assigns the grid pressure level (LEVP).
Assuming LEVP has been keyed in, it must be the same as the level of the grid just read from the gridfile (LEVEL), or GWVA will terminate. The value of LEVP will later determine whether the final wind data will be stored in the retrieval MD file (if LEVP is .NE. 0, data stored, and vice-versa). Following this, a check is performed in DO LOOP 14 to see if the keyed-in level is a mandatory level.

At this point, the lat/lon grid boundaries (LAN, LOW, LAS and LOE) and grid increment (INC) are determined, as well as the maximum and minimum latitude and longitude values for which winds can be computed from the height grid (LATMAX, LONMAX, LATMIN and LONMIN).

In the next important step, the existence of the retrieval MD file is checked via function MDFINFO, after which the keys for the file are read via function MDKEYS. Following a test designed to make sure data exists in the retrieval MD file, during which the row header of the retrieval file corresponding to row MDR is read, program GWVA enters DO LOOP 11. In this loop, the addresses within the keylist corresponding to such things as number of reports within a given retrieval MD file row, retrieval latitude, and retrieval longitude are determined. Some of this information is then used to assign a value to variable LAST.

Assuming LAST now has a value .NE. 0, variables DELT and NGR are set. If the user wishes to produce either isallobaric or ageostrophic winds, he/she should key in keyword parameters DELT and NGOLD with non-zero values, since both of these winds require changes of wind over a period of time or, in other words, another
earlier height grid. If indeed one of these wind options is desired, the earlier height grid is read by function IGGET ((IF DELT .NE. 0) IOK=IGGET(NGRF,NCR,2400,JHIT2,NR,NC,IGHD)). Finally, GWVA enters DO LOOP 200. The purpose of this large loop is to calculate the desired winds at each retrieval location within the grid boundaries, and plot the results if the plot option is in effect. After the retrieval report is read via function MDGET (IOK=MDGET(MDNO,MDR,N,IBUF)), GWVA enters DO LOOP 110. This loop finds the address within the keyword list which corresponds to surface pressure (to be used subsequently when accessing a given retrieval's surface pressure from the array IBUF), and also computes the value of variable NWIN, which will be used later near the end of GWVA to determine into which words of IBUF the final wind direction and speed will be placed. Note that if NWIN is never set .NE. 0 (i.e., no retrieval pressure level matches the grid pressure level), but the grid pressure level (LEVP) WAS keyed in, no wind will be calculated at that particular retrieval location. In addition, the winds will not be calculated if either the pressure level of the wind is greater than the retrieval surface pressure (i.e., the wind is below ground), or if the retrieval location is outside the previously-determined wind calculation boundaries for the grid (LATMAX, LONMAX, LATMIN, LONMIN).

Beyond this point in GWVA, the discussion will be divided into two categories: plot option and no plot option.

First, if the plot option is in effect (IPLT=0), GWVA calculates TV coordinates (raster (JP), pixel (IP)), first by
converting retrieval latitude/longitude to satellite coordinates (FLIN/FELE) via subroutine SATEAR, and then in turn converting these coordinates to the desired TV coordinates via subroutine SATTV. Following this, the desired wind is calculated at grid location (I,J) by subroutine ZWIND. Then, the wind barb itself is plotted on the video (TV) screen by subroutine BARB, after the wind barb's direction has been adjusted by DIRADJ (am assuming here that NSAT is .NE. 0). Finally, if a level was keyed in earlier as the second positional parameter (LEVP .NE. 0), the wind direction and speed are put into the retrieval MD file via function MDPUT (IK=MDPUT(MDNO,MDR,N,IBUF), after which DO LOOP 200 loops back to process another wind at a new retrieval location. On the other hand, if no level was keyed in, none of the wind results will be put into the retrieval MD file.

Second, if the plot option is NOT in effect, GWVA immediately calculates the wind at the grid location. Then, after skipping to statement 170 (remember, IPLT .NE. 0); the program proceeds the same as the case described above.
TAKE THIS ROUTE IF
A GRID PRESSURE LEVEL
WAS KEIED IN

LEVEL = LEVP

CALL
SDEST('GRID LEVEL
DOES NOT AGREE WITH
LEVEL KEIED IN...', LEVP)

END

I=I+1
I>10

DERTERINE IF GRID
PRESSURE LEVEL IS A
MANDATORY LEVEL

LEVEL = PRESS

NLV = I

GET HORIZONTAL
INCREMENT IN METERS

LATMAX = (LAM - 2*INC) +INC2 X10

DERTERINE LAT/LON BOUNDS
WITHIN HEIGHT GRID FOR
WIND CALCULATIONS
(BOUNDARIES ARE WITHIN
ACTUAL GRID BOUNDS)

LOWMAX = (LOW - 2*INC) +INC2 X10

LATMIN = (LAM + 2*INC) +INC2 X10

LCOMIN = (LOE + 2*INC) +INC2 X10

CHECK
EXISTENCE
OF RET.
MD FILE

MDINFO = MDINFO, MDNO, MBDLO

CALL SDIST
(CANNOT OPEN
MD FILE NO', MDNO)

IF MDINFO = φ

END

READY IN KEYS

READ KEYS FOR RETRIEVAL
MD FILE

TEST TO SEE IF DATA
EXISTS IN THE RET. MD FILE

READ RETRIEVAL MD FILE

READ KEYS FOR RETRIEVAL
MD FILE

READ TEST RECORD

I0K = MDGET(MDNO, NDR, M, IBUF)

READ TEST RECORD

READE RETRIEVAL MD FILE

REPORT

184
CALL ZWINO (JHT, J, I, JLAT, SPEED, DIREC, JHT, JHT2, JHT3)

CALCULATE WIND OF DESIRED TYPE

IF NSAT = 0
CALL DIRADJ (FLAT, FLOM, SATLOM, DIREC, ADJ)

PLOT WIND BARB

CALL BARB (ADJ, SPEED, XR, YR, COL, SZ)

CHECK HEIGHT LEVEL FOR WRITING WIND OB

IF SECONb POSITIONAL PARAMETER NOT KEPT IN (LEV=0), AS RESULT, THE FINAL WINDS WILL NOT BE WRITTEN TO THE RETRIEVAL MD FILE

170

"NO-PLT" ROUTE

ADJUST DIRECTION OR WIND SO PLOT WILL BE FROM TRUE DIRECTION ON VIDEO SCREEN

175

WRITE RETRIEVAL REPORT, UPDATED WITH WIND DIRECTION AND SPEED INTO RETRIEVAL MD FILE

IF IOK = MDPUT (MONO, MDL, N, ISUP) MD FILE

CALL SDEST ("TROUBLE WRITING DATA RECORD NO, I"), END OF DO LOOP 2φΦ

END

CALL ENOPLT

CALL SDEST ("DONE...", φ)

END

END OF DO LOOP 2φΦ
Subroutines used by GWVA:

1) INITPL-I
2) SDEST-I
3) NVINIT-I
4) SATEAR-I
5) SATT-V-I
6) ZWIN-D-I
7) DIRADJ-I
8) BARB-I
9) ENDPLT-I
10) DOPEN-I
11) DREAD-I
12) TVSAT-I
13) TSNIO-I
14) OUTINT-I
15) SRCH-I
16) ENCODE
17) ENCODX
18) ABORT
19) CONTNT
20) LTQ
21) ZECONV
22) MOVB
23) CLEANA
24) TQ
25) PRLINX
26) PRCLOS
27) PROPEm
28) PRPRPR
29) LOCK
30) PRWR
31) PRRD
32) UNLOCK
33) PRCL
34) POST
35) BLKA
36) DCLOSE
37) II
38) STC
39) DWRITE
40) MOVW
41) GETNAV
42) EPOCH
43) TQMES
44) ITOC
45) GETGAM
46) VASNAV
47) DDEST
48) CLEANW
49) EDDEST
50) MOVW
51) SATPOS
52) GETFRM

53) TRMNLI
54) GRADWI
55) PLOT
56) ENPT
58) PENADD
59) PACK
60) SENOVT
61) ATOE
62) TEKPUT
63) PAGE
64) PENMOV
65) BOX
66) PENBEG
67) WALK
UGVA

UGVA is used to update the first guess gridfile (NGFG) using VAS retrievals. It is a program that can be used in the event two or more sounder areas of different times have to be processed, because it allows the latter areas to use more up-to-date first guesses. For instance, let's assume we have two sounder areas at 1400 and 1600 Greenwich Mean Time (GMT), and a 1200 GMT first guess. The first guess would be used to process the 1400 GMT retrievals. Then, UGVA could be used to generate grids of 1400 GMT retrieval parameters by updating the original first guess grids with the 1400 GMT retrievals. The resultant grids (normally one of 1000 mb height, 15 of temperature and 6 of dewpoint) could then be used as a first guess for the 1600 GMT retrievals.

Essentially, as the reader will soon see, all UGVA does is run BNVA 22 consecutive times, using a given first guess grid for each execution.

After the first grid to be updated has been determined, the VASTEXT file has been opened and read, and the guess gridfile to be updated has been determined, UGVA enters DO LOOP 10, which encompasses the vast majority of the remainder of the program.

First, a given grid N is read from the guess gridfile via function IGGET. UGVA terminates if there is no such grid file. In addition, the program will move to read the next grid if grid N does not exist or is too big to be read. Assuming the grid is read with no problems, the gridded variable's name (ITVNAME) is found in one of four character lists (LCHAR, ICHAR, KCHAR and JCHAR), and the pressure level of the grid in question is checked
in DO LOOP 5 to see if it equals element J of the array IPRESS. UGVA will move to read the next grid if either the gridded variable's name or the pressure level of the grid cannot be found. Assuming that the name and pressure level of the grid ARE found, the grid header is stored in array IGHD, and word L of the array IGOT is set to 1 to prevent UGVA from generating any subsequent grids of the same parameter at the same level. After this, subroutines ENKODE and TQ display on the CRT such information as the name of the gridded variable (JCHAR(JB)), the grid's pressure level (ITLEVL), and the number of the grid within the guess gridfile that is to be updated (N). Grid N will serve as the guess grid in the forthcoming execution of program BNVA.

Next, in the most important step of UGVA, a call to subroutine JSQX results in the execution of program BNVA and the generation of a resultant grid of the specified retrieval parameter at level ITLEVL, using guess grid N. This updated guess grid will then be stored in the first guess gridfile in the first empty slot after O. Finally, variable NGOT is checked to see how many grids have been generated. If NGOT=22 (hence, 22 grids created), DO LOOP 10 is exited. As mentioned earlier, there will normally be 22 resultant grids, consisting of 15 grids of temperature, 6 grids of dewpoint and 1 grid of 1000 mb height.
READ NUMBER OF FIRST GRID TO BE UPDATED

OPEN AND READ VASTEXT FILE

GET NUMBER OF FIRST GUESS GRID FILE FROM VASTEXT FILE

DO LOOP 10

GET GUESS GRID

IOK = IGGET (NGFG, N, 3200, /GRID(1), NR, NC, IGHD)

ITVNAME = IGHDT(7)

ITLEVEL = IGHDT(18)

FIND GRIDDED VARIABLES NAME IN CHARACTER LISTS

CALL SDST ('UNABLE TO OPEN GRID FILE NO', NGFG)

END
Subroutines used by UGVA:

1) DOPEN-I
2) DREAD-I
3) ENKODE-I
4) TQ-I
5) JSQX-I
6) SDEST-I
7) ENCODE
8) ABORT
9) ENCODX
10) CONTNT
11) LTQ
12) ZECONV
13) MOVB
14) CLEANA
15) PRLINX
16) PRCLOS
17) PROPER
18) PPRPR
19) LOCK
20) PRWR
21) PRRD
22) UNLOCK
23) PRCl
24) POST
25) BLKA
26) ISQX
27) LWCLOS
28) MOVWC
29) LWMOP
30) TOKANL
31) SOSLED
32) PUC
33) WD
34) MOVCVW
35) LWPO
36) EDEST
37) MOVW
38) CLEANW
39) LWNEWF
40) LWSO
41) JMBWTF
CHAPTER 4

Supplementary VAS Retrieval Software

The software in this chapter should be considered as having an importance secondary to that in Chapter 3. These programs are used less frequently, but can on occasion still make a meaningful contribution to the retrieval process as a whole. The format used in describing the programs in this chapter is identical to that used in Chapter 3.
EXVA

EXVA is used to examine retrieval profiles in terms of temperature and differences in temperature between the retrieval and first guess profiles. The results for each retrieval examined are displayed on the CRT. One or more retrieval profiles can be examined and displayed on the CRT at one time.

Initially, the cursor coordinates (in lines and elements) are determined (CALL TVSAT(IFRM, IRAS, IPIC, IL, IE, JS, JD, JT)), and the VASTEXT file is opened and read. Then, after the navigation has been initialized, the retrieval MD file and row numbers are put, respectively, into the variables MDNO and MDR. These numbers are automatically taken from the VASTEXT file, and correspond to IDOC(40) and (41), respectively. Following this, if the retrieval MD file is successfully opened for reading, the keys for the file (schema VRET) are read, which allows EXVA to determine which words within the retrieval MD file output array (IOUT) for a given retrieval report contain latitude, longitude and surface temperature (NLAT, NLOX and NDX, respectively). The values for latitude, longitude and surface temperature will be stored in the same word locations for each retrieval report in the retrieval MD file.

After the existence of the retrieval MD file has been checked with function MDINFO, the row header for row MDR of the file is read, which gives, among other things, the number of retrieval reports contained within that row of the file (stored in variable MREC). In addition, the numbers of the first and last retrievals whose data is to be displayed, if more than one
retrieval report is desired, are read into variables N and M. Note also the determination of variable SLOP. This variable is used as a search radius when only one report at(near) the cursor location is to be displayed.

From this point on, the discussion of the remainder of EXVA will consist of three paragraphs. Each paragraph will discuss how EXVA operates for a given keyin. The first paragraph assumes the user has simply keyed in EXVA by itself. The second paragraph assumes that one retrieval number has been keyed in with EXVA, while the third discusses the case when EXVA is keyed in with two retrieval numbers as positional parameters.

Assuming no retrieval number or numbers have been keyed in (N, M .EQ. 0), only the retrieval report at(near) the cursor location will be presented. In this case, subroutine SATEAR calculates the latitude and longitude of the cursor. Then, DO LOOP 200 goes through each retrieval report in row MDR of the retrieval MD file, looking for any retrieval that is suitably close to the cursor location (note variables FCK and SLOP; SLOP=0.75 degrees, if it was not explicitly keyed in earlier). If no retrievals close enough are found, EXVA terminates. However, if a satisfactory retrieval IS found (FCK .LE. SLOP), its latitude, longitude and number within row MDR are printed on the CRT, as well as surface temperature, 850 mb temperature (T850), T700, T500, T400, T300, T250 and T200. In addition, the temperature differences between the retrieval and first guess for the same levels are also printed. This then concludes EXVA for the no retrieval number option.
If only one retrieval number has been keyed in (N .NE. 0, M=0), the retrieval report corresponding to the number N is read from the retrieval MD file (IOK=MDGET(MGNO, MDR, NN, IOUT)), and the same information as described above is printed on the CRT, which will then bring this particular option of EXVA to a close.

Finally, if two retrieval numbers have been keyed in, the data for all retrievals between and including the two numbers is printed on the CRT, provided of course the number of the retrieval to be displayed (N) does not exceed MREC and that there is a record at each report location in row MDR of the retrieval MD file.
EXVA

IHNL = IKWP ('HELP', 1, 0)

F

CALL SDST ('INUM-START')/('INUM-STOP')/('IDOC')

CALL SDST ('IF NO NUM, PULL UP RPT AT CURSOR LOCATION', 'IDOC')

END

CALL TKSAT (IFRM, IRAS, IPIC, IL, IE, JS, JD, VI)

READ IN DOCUMENTATION

LUCA(0) = 0

CALL TDIO(1, 1, 1, 1, IDOC)

OPEN, READ AND CLOSE VASTTEXT FILE

CALL DOPEN (IPFILE, LUN, LEN)

CALL DREAD (LUN, ITERM, IDOC)

CALL DCLOSE (LUN)

OPEN RETRIEVAL MD FILE FOR READ

RETRIEVAL MD FILE AND ROW NUMBERS SET

CALL NVUNIT (BETAIN, BETODT, INAV, PTIME)

MDNO = IDOC (48)

MUST CALL MDOPEN TO USE MDGET

CALL SDST ('CANNOT OPEN MD FILE... USE MDU SET', 'IDOC')

END

READ IN KEYS

SET UP INDICES
CALL OUTINT(KOUT)

CALL SDEST('NO SOUNDING AT THIS LOCATION...','

CALL OUTINT(KOUT)

END

CALL OUTINT(KOUT)

ICUR = 0

F

M = 0

F

N = N + 1

F

N > M

PRINT LAT, LON AND RETRIEVAL NUMBER ON CAT

CALL SDEST(7000, T850, T700, T500, T250, T200, 1, 1)

OUTPUT RETRIEVAL TEMPS. AND RETRIEVAL GUESSED TEMPS.
DIFFERENCES FOR THE LEVELS LISTED IN THE ABOVE CALL TO
"SDEST(SFC, 850, 200..."
QUIT IF DATA FROM ONLY ONE RETRIEVAL AT (NEAR) CURSOR IS BEING PLOTTED

QUIT WHEN N BECOMES > NO OR LAST RET. REPORT TO BE DISPLAYED (FOR EITHER ONE OR SEVERAL REPORTS)
RETURN TO DISPLAY NEXT REPORT IF MORE THAN ONE IS TO BE DISPLAYED
Subroutines used by EXVA:

1) TVSAT-I
2) DOPEN-I
3) DREAD-I
4) DCLOSE-I
5) NVINIT-I
6) TSNIO-I
7) SATEAR-I
8) SDEST-I
9) OUTINT-I
10) GETFRM
11) TRMNL
12) TQMES
13) BLKA
14) STC
15) ITOC
16) MOVV
17) LTQ
18) CLEANA
19) TQ
20) PRLINX
21) PRCLOS
22) PROOPEN
23) PRPRPR
24) LOCK
25) PRWR
26) PRRD
27) UNLOCK
28) PRCL
29) POST
30) ABORT
31) ENCODE
32) ENCODX
33) CONTNT
34) ZECONV
35) GETNAV
36) EPOCH
37) GETGAM
38) VASNAV
39) DDEST
40) MOVVCW
41) CLEANW
42) EDEST
43) MOVW
44) II
45) DWRITE
46) SATPOS
VDVA

The purpose of this program is to pick up VAS data according to the position of the cursor (or keyed in line and element) and the sonuder-area pointer (set previously by program VPVA). It is a program that is not vital to the generation of a retrieval or area of retrievals; rather, VDVA can be used to check an individual field of view to make sure no unreasonable values of brightness temperature, spin budget, radiance, etc. exist for any given band.

Eight different general types of keyin possibilities exist for VDVA, depending on which positional parameter(s) is(are) included in the keyin. For VDVA keyed in with one following positional parameter, the five possibilities are as follows: VDVA B (brightness temperature), VDVA R (radiance), VDVA F (filter number), VDVA S (spin budget) and VDVA N (navigation). In order to read the VAS data, all eight options require the cursor position to be known in satellite coordinates (line-IL, element-IE). Normally, the cursor position, in terms of satellite coordinates, will be determined via subroutine TVSAT. However, an option (#6) does exist to key in the line/element coordinates of the field of view to be examined via the second and third positional parameters in the program keyin (VDVA F 2000 9900, for example). If this option is taken, the call to TVSAT will not be executed, because MIN(2) will not be equal to 0 (see "IF(MIN(2).NE.0)GO TO 110").

For the default option (#7--VDVA keyed in by itself), the output consists of brightness temperatures for bands 1-12,
with the output for band 11, deleted due to excessive noise, consisting of six 9's. Other output for this option includes variable JD, which consists of a combination of the satellite number + year + Julian date, as well as cursor location data, local (or satellite--LZEN) and solar (SZEN) zenith angles, land elevation, and surface characteristic (0=ocean, 1=land). The output for option B (keyin of "VDVA B") consists only of the brightness temperature data for all twelve bands. The output for option N consists of variable JD, as well as the cursor location data, zenith angle data, land elevation, and surface characteristic. In other words, the output for option N consists of the default option information less the brightness temperature data. Option R returns radiance data only for the twelve bands.

Option F returns the filter (sensor) number being used for each band, while the last single positional parameter option, S, simply returns the spin budget for each band.

Finally, if the user keys in VDVA followed by one of the five explicit positional parameters described above (B, N, R, F or S), two zeroes (positional parameters 2 and 3), and then a "1" (#8--a keyin of "VDVA B 0 0 1," for example), the information noted above for options B, N, R, F or S, as well as further supplementary diagnostic information, is printed on the CRT. This diagnostic information always includes the sounder area directory and, in most cases, information pertaining to image to sounder-file (sounder area) coordinate transformations, raw data counts for each band, etc.
Subroutines used by VDVA:

1) IQ-I 53) LWOPEN
2) TVSAT-I 54) LWGET
3) VASDAT-I 55) SATEAR
4) SDEST-I 56) SATPOS
5) HRTOP0-I 57) ANGGET
6) OUTINT-I 58) SOLARP
7) PUTCHR 59) ANGLES
8) GETFRM 60) READOF
9) TRMLN 61) RBYTSX
10) TQMES 62) ZEROS
11) BLKA 63) RDTRK
12) STC 64) MOVCH
13) ITOC 65) LWPO
14) MOVW 66) LWNEWF
15) LTQ 67) LWSO
16) CLEANA 68) JMWBTF
17) TQ
18) PRLNIX
19) PRCLOS
20) PROPEN
21) PRPRPR
22) LOCK
23) PRWR
24) PRRD
25) UNLOCK
26) PRCL
27) POST
28) ABORT
29) MOVCH
30) NVINIT
31) GETNAV
32) EPOCH
33) GETGAM
34) VASNAV
35) DDEST
36) EDEST
37) MOVW
38) CLEANW
39) READD
40) II
41) PLNKIV
42) DOPEN
43) ENCODE
44) ENCODX
45) CONTNT
46) ZECOV
47) DREAD
48) DCLOSE
49) LWCL0S
50) MOVW
51) LWMP
52) OPNA
CHAPTER 5

Miscellaneous VAS Retrieval Software

The routines in this chapter are composed of a mixture of rarely (if ever) used retrieval support and deactivated retrieval-processing software. Since any or all of these programs may perhaps be rejuvenated at some future date, it is felt that they too should be included in the manual.

The deactivated software consists of programs SRAD, SRET and VTPW. SRAD and SRET together were used to process routine VAS retrievals up to approximately early April, 1984, while VTPW was used on an experimental basis early in 1984 to generate images of total precipitable water vapor from VAS radiance information (VTPZ performs this function now, in addition to doing the retrievals).
SRAD

This program prepares VAS large or small-detector (approximately 16 and 8 km resolution, respectively) brightness temperatures (TBs) for SRET, the VAS iterative retrieval program. Initially, before SRAD is keyed in, the cursor should be moved to a position near the northwest (NW) corner of the retrieval area the user wishes to process. The reason for this will become clear later. When the program is keyed in, the first positional parameter should be entered as "GO". Suitable keywords will then be entered after the "GO". (The option of running SRAD with the first positional parameter equal to "CLE" should be used only if the user wishes to update words 50-60 (IDOC(50-60)--bottom row of output as displayed by program LOVA) in the VASTEXT file. This option will rarely be used.) SRAD can be keyed in to process either an area of retrievals or one individual retrieval. This discussion of SRAD will focus on the processing of an area of retrievals, and will be concerned primarily with the "non-auto" option (keyword parameter AUTO not keyed in). Any deviations from this format will be duly noted. Note also that occasionally radiance and TBB will be interchanged. The two, of course, can be related through the Planck function.

Initially, the VASTEXT file and the row header of the retrieval MD file are read, and the first positional parameter is evaluated. Assuming the user has keyed in "GO" for this first parameter, the program jumps to statement 100 (IF CPP(1,') .NE.'CLE') GO TO 100), where the latitude/longitude bounds which determine whether a given retrieval box of VAS radiance
data can be used in the processing of the retrieval area (LN, LS, LW and LE) are set. Note that these values are derived from VASTEXT file values IDOC(25) and (26). Then, several other parameters, such as retrieval type (variable CTYP--a keyin of "TYP=N, C or B/no keyin" for N-star, clear or default retrievals, respectively), surface data option (NOSFC), retrieval box line and element spacings (in fields of view (FOVs)--INCR1,INCRE), retrieval box size (in FOVs--NBXS), etc. are set. Note that the retrieval box size can range from 3*3 to 11*11 FOV's in dimension. However, assuming the box size has NOT been keyed in via keyword parameter SIZ, the default size determined for large detector retrievals will be 5*5 FOVs, while the same for small detector retrievals will be 9*9 FOVs. After the retrieval box size has been determined, the debug option (KBUG) is set. If KBUG is set to 1, 2 or 3 (keyword parameter BUG=1,2 or 3 in keyin), messages detailing program status will be displayed at the corresponding location (either the CRT, local printer or system printer, respectively). Then, the plot option (IPLT) and variables MDNG (retrieval guess MD file) and MDNR (retrieval MD file) are given values. Note that MDNG and MDNR will also get their values from the VASTEXT file.

At this point, the cursor TV coordinates (raster(INRAS), pixel(INPIC)) are accessed from User Common via function LUC, after which SRAD jumps to statement 112 (remember, SRAD is in the non-auto mode), where the initial and last line and element positions for the area of retrievals to be processed is
determined. The last line and element positions (SE corner of image—LLINE/LELEM) will usually be keyed in using keyword parameter END as part of the original program keyin. The initial positions (NW corner of image—ILINE/IELEM) normally will NOT be keyed in; as a result, variables ILINE and IELEM will both initially assume the value 0 (DATA statement near code line 70) and will subsequently be determined via subroutine TVSAT. (The user should be able to see now why the cursor had to be positioned near the NW corner of the desired retrieval area before SRAD was executed.) Before the call to TVSAT, however, a call to subroutine VRTIO results in an update to the VASTEXT file, based on the changes made to this point in the array IDOC.

As an aside, assuming for a moment that SRAD is in the auto mode (keyword parameter AUTO keyed in with a value of 1), variables LLINE, LELEM, ILINE and IELEM are gleaned from the VASTEXT file. These values were calculated originally in program IDVA. Then, after the call to VRTIO, SRAD jumps to statement 218, and then to statement 220.

Returning now to the non-auto mode discussion, the single retrieval case will be mentioned briefly. SRAD will perform only one retrieval at the cursor location if the last line and element positions have not been keyed in (LLINE, LELEM = 0). If this is the case, ILINE and IELEM will again be determined by TVSAT, after which the program will set the SE and NW line and element positions equal to each other.

Finally, the discussion returns to the original non-auto mode—retrieval area discussion. After TVSAT has been called
(just before statement 218), SRAD jumps to statement 220 (IF (LLINE*LELEM.NE.0) GO TO 220). Immediately after this statement, subroutine VASDAT accesses navigation data, including information such as image line and element resolution. Then, after the retrieval box spacings in satellite coordinates (lines (y-direction--INCIL), elements (x-direction--INCIE)) have been determined, subroutine GETFRM returns a directory which contains image/image frame information (including the image magnification), after which SRAD is ready to begin processing the actual TBB data. (For a schematic showing the layout of a typical retrieval area, see Diagram 2 immediately following this discussion.)

The processing is done retrieval line by retrieval line within DO LOOP 2400, moving successively left to right within each line retrieval box by retrieval box (see statement 230: IEP=IEP+INCIE). Assuming IEP .LE. LELEM, the code up to the comments preceding statement 305 (DO LOOPs 300 and 280, primarily) serves to initialize several of the arrays and variables to be used in the subsequent TBB processing.

Beginning with statement 305, SRAD enters an implicit DO LOOP which runs for approximately the next 45 lines, the main purpose of which is to access VAS brightness temperature data for all available bands for each FOV within a given retrieval box. Subroutine VASDAT does the actual data gathering. If the first FOV's TBB data, which will be at the retrieval box center, is being accessed (N=1), some tests are performed to make sure VASDAT is accessing data from a reasonable location. For
instance, if the latitude of this FOV is north or south of the
previously-set retrieval area latitude boundaries (LN,LS), the
program will move to the next retrieval line within the image to
attempt further processing. Other conditions causing problems
include longitude greater than the western boundary (in which
case SRAD steps one-half retrieval box spacing to the right and
tries again), or longitude less than the eastern boundary (in
which case SRAD jumps to statement 2390, updates the retrieval MD
file row header and the VASTEXT file with the total number of
retrievals completed so far (LASRET) via subroutine VRTIO, and
again moves to the next retrieval line within the image for
further TBB processing). SRAD will also move one-half retrieval
box spacing to the right and attempt to access another set of TBB
data if the satellite zenith angle (VZEN) is .GT. 60 degrees and
LBEG .EQ. 0. LBEG will equal 0 until the TBB data for the first
retrieval box in a given retrieval line has been accessed.
Assuming there are no problems gathering the TBB data for a given
FOV, the TBB information accessed will be collected for each band
in the FOV and stored in the array RADS (N,I) in DO LOOP 400. DO
LOOP 400 also updates a given element of the TBB array VDAT with
the value VMISG if any gross TBB anomalies exist, in addition to
tabulating the total number of FOV's in a given retrieval box
(MSAM). After DO LOOP 400, variables SELEV, SLAT and SLON are
updated with the values for the FOV being processed. The implicit
DO LOOP will be exited after variable IBOX has been set to 3 in
VASDAT, which will occur when all the radiance information for
the retrieval box in question has been gathered.
Following the implicit DO LOOP, and assuming MSAM .GT. 0, the average retrieval box surface elevation, latitude and longitude are stored in the variables SELEV, SLAT and SLON, respectively. Then, a call to subroutine SURGES returns surface information, including variable ITSFC (surface temperature), which is used to determine variable TSTA. Following this, variable TMN (used to screen cloud-contaminated FOVs in a given retrieval box) is determined, and DO LOOPs 580 and 560 evaluate all the FOV's in the retrieval box, summing up for each band the number of FOVs with TBB data, and the TBBs themselves, in the arrays NSAM (or IUSE) and AVG, respectively.

After DO LOOP 580 has been exited, a quality control check involving the band 8 (window) brightness temperature data is done to see if there are at least 3 TBB reports in the retrieval box for this band. If the check fails, the program will step one-half retrieval box spacing to the right and attempt another retrieval. Otherwise, DO LOOP 590 will set IUSE(K)=0 for any band K other than band 8 which does not have at least 3 FOVs in the retrieval box with TBB observations.

Up to and including DO LOOP 590, SRAD has executed the same for all three retrieval type options (default, N-star or clear.) From this point on, the function of SRAD depends upon which type of retrieval is being done. As a result, separate discussions will be given for each retrieval type.

If the retrieval type is N-star (used under partly cloudy conditions), SRAD skips to statement 770, where clear column radiances for the retrieval box are inferred mathematically in
each band from the mixture of clear, partly cloudy and cloudy radiances by subroutine NSTAR (MacMillin, 1978). Failure in NSTAR (IFAIL.NE.0) results in a new retrieval attempt one-half retrieval box spacing to the right of the failed attempt.

Otherwise, if subroutine NSTAR is successful, SRAD enters DO LOOP 830, in which the expected error of the brightness temperature observations for each band 1-12 (the array EX) are computed. After this DO LOOP, the surface skin temperature (TS) is calculated. Then, the output buffer (the array IRET--note equivalence of to the array IRETID as stated back near code line 55) containing, among other things, the final brightness temperature and expected error of the brightness temperature observations for each band (DO LOOP 2203), is filled. Following this, a call to subroutine VRTIO (CALL VRTIO(IRET,LASRET,1)) stores IRET in row MDRR, column LASRET of the retrieval MD file. Then, if the plot option is in effect (IPLT .NE. 0), the retrieval box band 8 TBB is plotted on the video screen at the retrieval location in KOLOR=3 (normally yellow) via subroutine VASDIG.

Finally, SRAD jumps back to statement 230 and increments one retrieval box spacing to the right to process another retrieval box, unless one of three things happens. First, if the number of the just-completed retrieval (LASRET) equals the maximum number of retrievals allowed (MAXRET), SRAD jumps to statement 2390, where the retrieval MD file row header is rewritten, the VASTEXT file is updated and, following a jump to statement 2500, the program terminates. Second, if the process of incrementing to the
right after a successful retrieval causes variable IEP to exceed LELEM (the rightmost element coordinate (x-direction) within the retrieval area for which retrievals can be processed), SRAD jumps to statement 2385, where the row header and VASTEXT file are again rewritten and updated, respectively, via VRTIO, after which SRAD moves to the next retrieval line to process that line's TBB data. For the third and final case, if the previous retrieval was the last one possible in the sounder area, SRAD concludes in a similar fashion to the second case above, except that instead of returning back to the beginning of DO LOOP 2400, SRAD exits 2400 and terminates. This third case concludes the discussion of SRAD in the N-STAR mode.

Returning now to the end of DO LOOP 590, if the retrieval type is clear or default, the maximum brightness temperature and the FOV in which the maximum brightness temperature occurs for a given retrieval box is computed for bands 6 and 8 in DO LOOP 600. Variables JSAV6 and JSAV store the FOV number(s) in which the maximum band 6 and 8 TBBs occur. After DO LOOP 600, if the maximum band 8 TBB is less than the previously-calculated threshold temperature TMN, or if either JSAV or JSAV6 = 0, SRAD moves one-half retrieval box spacing to the right and attempts a new retrieval.

Assuming that the threshold test passes, and that JSAV6/JSAV both have values, SRAD sets up "screening variables" in bands 4,5,6,8 and 9 (RLO4,RLO5, etc.) to delete cloud contaminated TBBs in bands 3,4,5,6,7,8,9,10 and 12 for all the FOVs in a given retrieval box. The actual screening is then done in DO LOOP 720.
Within this loop, different elements of the array LC are set to 0 to indicate cloud contamination in various VAS bands, depending on how the TBB's in bands 4, 5, 6 and 8 for a given FOV compare to the screening variables RLO4, RLO5, RLO6, RLO8 and RLO9. In addition, as part of DO LOOP 720, DO LOOP 700 recalculates the arrays NSAM and AVG with the clearest screened radiance information. Following DO LOOP 720, DO LOOP 760 calculates the average retrieval box brightness temperature for each band, with the results being placed in the array VDAT.

At this point, if the retrieval type is clear (FORCE=10. see statement 105), or if the retrieval type is default (FORCE=0.) and there are 5 or more clear band 7 FOVs, SRAD jumps to statement 790, where the expected error of the brightness temperature observations for each band are computed, after which the surface skin temperature is calculated via function VSKINT. Finally, the output buffer and plot option sequence is completed in the same fashion as the previously-discussed N-star retrieval path, except that the band 8 brightness temperature is plotted in KOLOR=2 (normally blue-green) instead of KOLOR=3.

However, if the retrieval type is default and there are LESS THAN five clear band 7 FOVs, the retrieval path will hereafter follow the previously-described N-STAR retrieval path. In other words, the default retrieval path is identical to the clear path, as long as there are at least 5 clear band 7 FOVs within the retrieval box, but default retrievals will become N-star retrievals if there are .LT. 5 clear band 7 FOVs.

This concludes the discussion of program SRAD. A summary of
DIAGRAM 2
TYPICAL RETRIEVAL AREAS

small detector

SATELLITE COORDINATE ELEMENTS (INCREASING)

SATELLITE COORDINATE LINES (INCREASING)

(RETRIEVAL LINE SPACING) INCIL

INCIL

1 FOV (20.8 x 20.8 km for small detector data)

4 SCAN LINES OF DATA

4 SCAN LINES OF NO S.D. DATA

1 SCAN LINE OF DATA

K-INCIL (RETRIEVAL BOX SPACING)

INCIL

NEXT RETRIEVAL LINE

NOTES:
1) DIMENSION OF RETRIEVAL BOX = 9 x 9 FOVs (NOT 11 x 11 AS IN ITPR AND VTPU)
2) THIS RETRIEVAL AREA HAS 4 RETRIEVAL BOXES PER RETRIEVAL LINE, WITH EACH RETRIEVAL BOX CONTAINING 81 FOVS. FOR THE FIRST RETRIEVAL LINE, THERE ARE 45 FOVS WHICH HAVE SMALL DETECTOR DATA (4 x 9 + 1 x 9), MEANING THERE ARE 36 FOVS WHICH HAVE NO SMALL DETECTOR DATA. FOR THE NEXT RETRIEVAL LINE, THE NUMBERS ARE AGAIN 45 AND 36, RESPECTIVELY. IT IS ONLY POSSIBLE TO HAVE EITHER 4 OR 5 SCAN LINES OF VAS DATA WITHIN A GIVEN 9 x 9 RETRIEVAL BOX.

large detector

LINES

1 FOV (26.6 x 26.6 km for large detector data)

1 SCAN LINE OF DATA

K-INCIL

INCIL

NEXT RETRIEVAL LINE

NOTES:
1) DIMENSION OF RETRIEVAL BOX = 5 x 5 FOVs
2) THIS RETRIEVAL AREA HAS 4 RETRIEVAL BOXES PER RETRIEVAL LINE, WITH EACH RETRIEVAL BOX CONTAINING 25 FOVS, ALL OF WHICH CONTAIN VAS 788 DATA.
SRAD

- EQUIVALENCE TO IRET(ROWLEN-1)
- ABOVE TO FACILITATE CHANGING LENGTH OF ROW HEADER

DATA VMASS/999999./NSMAX/121/

DATA MAAA/699/

- FOLLOWING IS PROGRAM VERSION DATE... KEEP CURRENT

CALL (CALDAY(IV,EV; IVM

IVD, IVMO))

CALL ENKODE("132X,T1,"BEGIN SRAD, VERSION OF", T2,
1X, A4, I2/), LOUN, IVD, IVMO, IIVY)

READ RETRIEVAL MD FILE
HOW HEADER, IN ADDITION TO THE VASTEXT FILE

CALL VARIO(IRET, &P, &P)

CALL MINDATE(IDOC, &P, IFILNM)

- ABOVE SETS UP FILE NAME TO FORCE OUTPUT AT END OF EACH LINE

CALL VARIO(IRET, &P, &P)

END

SET LAT/LOH BOUNDS

LN=LLNW/1000

LS=LLSE/1000

LN=MOD(LLNW, 1000)

LE=MOD(LLSE, 1000)

- ABOVE WILL LIMIT AREA OF RETRIEvals

- TEST FOR CALIBRATION RUN

CTYP = CRWP("TYP, 1")

CTYP = 'C'

SET FORCE FLAG FOR CLEAR RETRIEVALS SO THEY WILL NOT BECOME N-STAR RETRIEVALS LATER DURING SAMP-7 CHECK (SEE CODE IMMEDIATELY PRECEDING IBM LOOP 78)
SET RETRIEVAL BOX LINE AND ELEMENT SPACING, AS WELL AS RETRIEVAL BOX SIZE

SET PLOT OPTION

GET CURSOR LOCATION IN TV COORDINATES (MASTER, PIXEL)

DETERMINE INITIAL AND LAST LINE AND ELEMENT COORDINATES FOR AREA OF RETRIEVALS TO BE PROCESSED

UPDATE THE VASTEXT FILE

IF FIRST POSITIONAL PARAMETER ≠ "GO", TERMINATE SRAO
1) SET UP SCREENING VARIABLES ALH, ALD, etc.
2) DO LOOP 717: SCREEN TBB VALUES IN BANDS 3, 4, 5, 6, 7, 8, 9, 10, and 12 FOR ALL RETRIEVAL BOX FOVS.
3) AS PART OF 717, DO LOOP 718: RECALCULATE THE ARRAYS NSAM AND AVG WITH THE SCREENED RADIANCE INFORMATION.
4) DO LOOP 719 FOR EACH BAND, CALCULATE AVERAGE RETRIEVAL BOX TBB.

CLEAR RETRIEVALS AND DEFAULT RETRIEVALS WITH ≤ 5 CLEAR BAND 7 FOVS TAKE THIS ROUTE.
<table>
<thead>
<tr>
<th>Subroutines used by SRAD:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1) CALDAY-I</td>
<td>53) GETNAV</td>
</tr>
<tr>
<td>2) ENKODE-I</td>
<td>54) EPOCH</td>
</tr>
<tr>
<td>3) VRTIO-I</td>
<td>55) GETGAM</td>
</tr>
<tr>
<td>4) MDNAME-I</td>
<td>56) VASNAV</td>
</tr>
<tr>
<td>5) TQSET-I</td>
<td>57) DDEST</td>
</tr>
<tr>
<td>6) GETDAY-I</td>
<td>58) READD</td>
</tr>
<tr>
<td>7) TVSAT-I</td>
<td>59) OUTINT</td>
</tr>
<tr>
<td>8) VASDAT-I</td>
<td>60) PLNKLV</td>
</tr>
<tr>
<td>9) GETFRM-I</td>
<td>61) OPNA</td>
</tr>
<tr>
<td>10) SDEST-I</td>
<td>62) LWOPEN</td>
</tr>
<tr>
<td>11) SURGES-I</td>
<td>63) LWGET</td>
</tr>
<tr>
<td>12) NSTAR-I</td>
<td>64) SATEAR</td>
</tr>
<tr>
<td>13) SATTV-I</td>
<td>65) SATPOS</td>
</tr>
<tr>
<td>14) VASDIG-I</td>
<td>66) ANGGET</td>
</tr>
<tr>
<td>15) ENCODX</td>
<td>67) SOLARP</td>
</tr>
<tr>
<td>16) CONTNT</td>
<td>68) ANGLES</td>
</tr>
<tr>
<td>17) LTQ</td>
<td>69) READOF</td>
</tr>
<tr>
<td>18) ZECONV</td>
<td>70) RBYTSX</td>
</tr>
<tr>
<td>19) MOVB</td>
<td>71) ZEROS</td>
</tr>
<tr>
<td>20) CLEANA</td>
<td>72) RDRTRK</td>
</tr>
<tr>
<td>21) TQ</td>
<td>73) MOVC</td>
</tr>
<tr>
<td>22) PRLNX</td>
<td>74) IGNAME</td>
</tr>
<tr>
<td>23) PRCLS</td>
<td>75) RORDER</td>
</tr>
<tr>
<td>24) PROPER</td>
<td>76) FILCLR</td>
</tr>
<tr>
<td>25) PRPRPR</td>
<td>77) FILCLD</td>
</tr>
<tr>
<td>26) LOCK</td>
<td>78) INITPL</td>
</tr>
<tr>
<td>27) PRWR</td>
<td>79) QCDASH</td>
</tr>
<tr>
<td>28) PRRD</td>
<td>80) DSHOFF</td>
</tr>
<tr>
<td>29) UNLOCK</td>
<td>81) PLOT</td>
</tr>
<tr>
<td>30) PRCL</td>
<td>82) DSHON</td>
</tr>
<tr>
<td>31) POST</td>
<td>83) ENDFLT</td>
</tr>
<tr>
<td>32) BLKA</td>
<td>84) ENPT</td>
</tr>
<tr>
<td>33) ABORT</td>
<td>85) PENADD</td>
</tr>
<tr>
<td>34) DOPEN</td>
<td>86) PACK</td>
</tr>
<tr>
<td>35) ENCODE</td>
<td>87) SENOUT</td>
</tr>
<tr>
<td>36) DREAD</td>
<td>88) ATOE</td>
</tr>
<tr>
<td>37) DCLOSE</td>
<td>89) TEKPUT</td>
</tr>
<tr>
<td>38) LWCLOSE</td>
<td>90) LWNEWF</td>
</tr>
<tr>
<td>39) MOVWC</td>
<td>91) LWSO</td>
</tr>
<tr>
<td>40) LWMOP</td>
<td>92) PAGE</td>
</tr>
<tr>
<td>41) LWPO</td>
<td>93) PENMOV</td>
</tr>
<tr>
<td>42) EDEST</td>
<td>94) BOX</td>
</tr>
<tr>
<td>43) MOVW</td>
<td>95) PENBEG</td>
</tr>
<tr>
<td>44) MOVWC</td>
<td>96) WALK</td>
</tr>
<tr>
<td>45) CLEANW</td>
<td>97) JMBWTF</td>
</tr>
<tr>
<td>46) II</td>
<td>98) GETSFV</td>
</tr>
<tr>
<td>47) STC</td>
<td>99) VROPO</td>
</tr>
<tr>
<td>48) DWRITE</td>
<td></td>
</tr>
<tr>
<td>49) TRMNL</td>
<td></td>
</tr>
<tr>
<td>50) TQMES</td>
<td></td>
</tr>
<tr>
<td>51) ITOC</td>
<td></td>
</tr>
<tr>
<td>52) NVINIT</td>
<td></td>
</tr>
</tbody>
</table>
SRET uses the VAS brightness temperature (TBB) measurements prepared by SRAD and retrieves vertical profiles of temperature and moisture. The profiles are retrieved using the iterative retrieval method described in Smith (1983).

Initially, after the program version date has been evaluated and displayed on the CRT via subroutines CALDAY and ENKODE, and the VASTEXT file and retrieval MD file row header have been read via subroutine VRTIO, important parameters such as the type of guess (CGES -- statement 10), surface data analysis option (NOSFC), and retrieval type (CTYP -- forced ("F"), default (no keyin or "B"), clear ("C"), or N-star ("N")) are determined. Note that VASTEXT file defaults IDOC(51) and (54), respectively, will be used for the guess type and surface analysis option if keyword parameters GSS and SFC are not entered as part of the keyin. This will result in a grid first guess, since IDOC(51) was set to 0 previously by program IDVA. In this discussion, I will concern myself mainly with how SRET runs in the "default" mode, since this is the retrieval type used most often, and I will also assume that the surface analysis option is being used. Other parameters determined during the pre-retrieval stage include the debug option (LCRT/KBUG) and the plot option (IPLT). Note the call to TQSET after LCRT has been set. TQSET will set the debug information output destination to 1 (CRT), 2 (local printer) or 3 (system printer), depending on the value of LCRT. Then, assuming a plot of the retrieval results is desired (plot will be of retrieved total-totals index), the cursor position in satellite
coordinates (line (IL), element (IE)) is determined by subroutine
TVSAT, the navigation file is opened via subroutine NVINIT, and
subroutine GETFRM returns a 64-word directory pertaining to the
VAS image, which includes things such as image frame number,
image magnification, etc. Finally, the guess and retrieval MD
file numbers (MDNG and MDNR, respectively) are gleaned from the
VASTEXT file (IDOC(38) and (40), respectively), the
temperature/moisture profile enhancement option (IENH) is set,
and the numbers of the first and last retrieval(s) to be
performed (NB and LASRET, respectively) are determined. The
number of the last retrieval will be either the keyed-in second
positional parameter value, or the number of reports indicated in
the retrieval MD file row header (IRETD(3)).

At this point, the main retrieval DO LOOP (2400) is entered.
This loop runs from the first through the last retrieval. For
each retrieval, the data prepared by SRAD is read in by
subroutine VRTIO, including the expected error of the brightness
temperature observations, as well as the averaged brightness
temperatures themselves, both for each VAS band. Then, assuming
the user mod flag .LT. 0 (retrieval not done yet), DO LOOP 700
stores these values in the arrays EX and VDAT, respectively. In
addition, the array VRAD stores computed Planck radiances for
each band. Note also the array IUSE. This array contains "use"
flags which indicate whether a given VAS band should be used in
the retrieval (a value of 1) or not used because it is missing or
has been intentionally set aside (a value of 0).

Now, after other quantities, such as satellite zenith angle
(VZEN), retrieval latitude/longitude (VLAT, VLON), the variable NOGO, etc. are set, the guess temperature and mixing ratio profiles for the retrieval are accessed via subroutine GESPRO. Assuming the guess is obtained without any problems (TGES(1) is NOT .LT. 0), if the guess surface temperature (TSTA) is more than 15 degrees warmer than variable TCK (approximately equal to the surface skin temperature as measured by the VAS), cloud contamination is indicated. If this is the case, the retrieval fails, and SRET jumps to statement 2200, where IRETD(3) (the user mod flag) is set to 7777. Then, a jump is taken to statement 2360 (remember, NOGO=1), where the array IRET (containing essentially only guess profiles) is put back into the retrieval MD file via subroutine VRTIO, and SRET moves on to process the next retrieval. Otherwise, assuming the cloud contamination test passes, the first pressure level below the surface is calculated in DO LOOP 400 (variable IS...maximum value = 40), and the lower portion of the guess temperature profile is adjusted to take into account the surface temperature (provided pressure level IS is .GE. 851 mb for a grid or profile first guess, and .GE. 501 mb for a climatology first guess (IGES=1)). Then, after some guess temperature and dewpoint data is prepared for the retrieval MD file in the arrays TGS and DGS, the guess information to this point (if KBUG is .GT. 0) is displayed on a given display device (=1 for CRT, 2 for local printer, 3 for system printer) via subroutine ENKODE. At this point, the surface dewpoint, dewpoint depression, pressure and mixing ratio are determined and (or) put into the variables TDSTA, DD, PSTA and WSTA, respectively.
The following 25 lines or so of code set a cloud flag (IFLG) either to 99 or 0, based on tests involving type of retrieval first guess, surface option, band 12 and 8 TBBs, surface skin temperature (TS), etc. The tests are used to determine if a cloud contamination problem exists for a given retrieval. If no cloud contamination problem is found, IFLG will be set to 0, while a cloud contamination problem will cause IFLG to assume a value of 99. If the flag is set to 99, the surface skin temperature calculated by SRAD (TS) will not be believed, and the temperature enhancement will not be done during the subsequent retrieval process. One of the tests, which is done only if the band 12 TBB value is to be used for the retrieval (IUSE(12) =1), compares the band 12 TBB value, which can be contaminated by excessive reflected sunlight from scattered cloud tops, with the surface skin temperature. If the band 12 TBB is more than 5 degrees Celsius warmer than TS, SRET will jump to 660, IFLG will be henceforth equal to 99, and the retrieval will be classified as overcast. A subsequent test compares the surface skin temperature to a threshold variable TMN. If the skin temperature is too cold, again indicating a cloud problem, IFLG remains equal to 99. If all the tests pass, and assuming the retrieval type is neither clear nor overcast (remember, this discussion deals with retrievals of type default), IFLG is set to 0, and the retrieval will subsequently be classified as clear.

Following the cloud contamination tests, and for cases where IFLG = 99, the pressure level immediately below the first level colder than the surface skin temperature (LS) is calculated
in DO LOOP 920. Then, statement 980 (NLS=MIN0(IS,LS)) calculates variable NLS, which is used to define the lower boundary level for calculating radiance via the radiative transfer equation (RTE) in the following relative humidity (RH) calculations (DO LOOP 1100). In summary, for cloudy cases where IFLG = 99, NLS will be the minimum of the first level below the surface (IS) and the just calculated variable LS (result will usually be LS, the cloud tops), while for clear cases where IFLG = 0, NLS will be the minimum of the first level below the surface and the number 40 (LS set equal to NL=40 just before DO LOOP 920).

The RH computations in DO LOOP 1100 involve only bands 7, 9 and 10, which are the VAS bands affected primarily by water vapor absorption. Before DO LOOP 1100 is entered, however, a mixing ratio array (S) is partially filled, resulting in values for the upper 25 retrieval levels of .001 g/kg. The subsequent net result of DO LOOP 1100 is 3 RH values based on bands 7, 9 and 10 that are assumed to be valid at 800, 600 and 300 mb, respectively, and represent layer RH values for the layers 1000-800, 800-600 and 600-300 mb, respectively. The RH values are stored in the array IHUM. In addition, the final value returned for each band layer will be between (or equal to) .10 and 1 (10-100%).

In the code following DO LOOP 1100, but preceding DO LOOP 1300, assuming IHUM(3) .NE. MISC, several miscellaneous operations are performed, affecting variables such as IHUM(1), IHUM(2) and RHS (surface relative humidity (RH)). Then, DO LOOP 1180 fills the array HUM with the final IHUM values.

At this point, DO LOOP 1300 is entered, in which the
nearly-completed retrieval mixing ratio profile (the mixing ratio array evaluated here (w) is updated during the forthcoming temperature retrieval section) is determined from a linear interpolation involving the 3 layer-RH values evaluated earlier, the surface RH value, and saturated mixing ratio values calculated from the guess temperature profile. After this loop has been completed, the transmittances for bands 7-10 are determined via subroutine VASTAU, after which the total precipitable water vapor (based on the nearly-completed retrieval moisture profile) is calculated by subroutine PRECW.

Now, SRET begins the temperature profile retrieval process. First, the number of possible temperature profile iterations (ITLIM) is set. The default number of iterations are 3 for the grid or profile first guesses, and 10 for the climatology first guess. Following this, the variables IPASS and ITER are initialized to 0. IPASS will be used to keep track of the number of ITERATIVE PASSES, while ITER will keep track of the number of temperature iterations WITHIN A GIVEN ITERATIVE PASS. Then, DO LOOP 1380 sets the values of variables KC1 and KC2, depending on whether a given band 1-6 (CO2 absorption bands) is to be used for the forthcoming temperature retrieval (note the value of element I or J of the array IUSE, and remember that a 0 indicates missing data or that the band has been intentionally set aside). These variables will form the loop bounds for DO LOOP 1520. Following 1380, since ICB will almost always be .NE. 1, a test will be conducted at statement 1385. In this test, if the lower bounds for the CO2 bands being used has been set .GE. 3, and the upper
bounds has been set .LE. 3, SRET will jump to statement 2200, where the user mod flag (IRETD(3)) will be set to 7777 (meaning "edit the report from the retrieval MD file") and, since NOGO is .NE. 0, a jump to 2360 will be taken, where the incomplete, non-retrieved report will be placed into the retrieval MD file via subroutine VRT10. Then, DO LOOP 2400 will attempt to process the next retrieval.

As a side note, from this point (statement 1390) all the way to statement 1951, the reader must pay strict attention to the current value of the variables ITER and IPASS. This is so because this section of code is somewhat complex, and constant familiarity with both of these variables is a must for a fuller understanding of the code. In addition, the user should concurrently follow the outline immediately after this discussion (labelled "SRET SUMMARY FROM STATEMENT 1390 TO STATEMENT 1951"), as it should help settle any confusion he/she may have. This outline summarizes the different possible paths SRET may take in the code between statements 1390 and 1951.

Returning now to the main discussion, assuming the test at 1385 passes, SRET enters DO LOOP 1520. For the first major step in this loop, the transmittances for all CO2 bands are determined through all 40 retrieval levels. The first time through the loop, variable IA, the lowest pressure level to which radiative transfer calculations will be done, is calculated. For clear retrievals (IFLG .EQ. 0), this level will simply be equal to IS, the first pressure level below the surface. For cloudy retrievals, however, DO LOOP 1463 re-evaluates IA, assigning it
the minimum of 2 values: the level immediately below the first level colder than the cloud tops, and the level IS. Note that IA is only determined once per retrieval (INIT set to 1, so SRET will henceforth jump to 1470).

After IA has been set, subroutine VASRTE calculates the radiance for a given band, also returning quantities such as the TBB calculated from the guess profile for that band, as well as the derivative quantities needed to calculate the weighting functions at each retrieval level for a given band. The weighting function calculations are then carried out in DO LOOP 1480. The final step in loop 1520 involves the calculation of the difference between the satellite-measured brightness temperature (VDAT(ICV)) and the brightness temperature calculated from the guess profile (TBB) (result known as the residual and stored in the array DELT). This information will eventually be used when evaluating the iterative form of the radiative transfer equation used to retrieve the temperature profile. Note that DELT values will NOT be calculated if either IUSE or KUSE(KC) equals 0.

Immediately following DO LOOP 1520, a data quality check is performed (only for the first temperature iteration, when ITER = 0, and for clear retrieval (IFLG=0) cases in which bands 5 and 6 are being used) to determine if there are problems with band 6. (As a side note, it was found empirically that the residual value for band 6 was occasionally erratic, compared to the band 4 and 5 residuals. Normally, the residuals were supposed to behave in a smoothly monotonic fashion (smoothly increasing/decreasing successively through bands 4-6-5), but they were found to
periodically deviate from this behavior. Consequently, this test was included to screen out the band 6 data when it was found to have problems.) The test involves several of the above-mentioned DELT values (AB5, AB6 and AB4). If AB6 .GE. AB5 and AB4, band 6 will no longer be used in the retrieval (IUSE(6)=0). After this test has been completed, another test evaluates the value of the variable CCK (DELT(5)-DELT(6)). If the absolute value of this variable is .GT. 2.5, and variable RHCK .GT. 0.95, a cloud problem is indicated, and IFLG is reset to 1, indicating a cloudy retrieval.

At this point (statement 1540), band 6 (weighting function peak at approximately 850 mb) is deleted from the retrieval process if the retrieval surface pressure is .LE. 850 mb. The subsequent code before DO LOOP 1560 sets aside one band (either 5 or 6) for later use in determining whether the retrieval has been successful, and also evaluates different cases for skipping DO LOOP 1560, including declared clear retrieval type (CTYP=C), no surface option-climatology first guess, or IFLG equal to 0 (default retrieval type determined as being clear).

Within DO LOOP 1560, suspected cloudy retrievals are checked, and those bands whose observed TBB values are more than 0.25 degrees Celsius cooler than the TBB derived from the guess temperature profile are deleted from the retrieval (KUSE(J)=0). In addition, note how the variable ITYP is set to overcast (23) if any band is found to be cloud-contaminated. ITYP is a retrieval-type flag which is stored in each retrieval report, and was determined initially in program SRAD. After DO LOOP 1560 has
been completed, a check is performed to see if band 4 (weighting function peak at approximately 450 mb) was deleted from the retrieval process due to cloud contamination (KUSE(4).EQ.0). If it was deleted, no retrieval is performed, and SRET jumps to statement 2200, where the same steps as discussed previously (approximately 3 pages back) are repeated.

Assuming the band 4 data is acceptable (KUSE(4).EQ.1), SRET moves to statement 1670 via an intermediate jump to statement 1662. After this, the program enters DO LOOP 1720, whose purpose is to retrieve the actual temperature profile by correcting the guess profile with the measured VAS radiances in each band. Within DO LOOP 1720, DO LOOP 1700 calculates the numerator and denominator of the temperature correction term to be added to the guess temperature at a given pressure level. Following DO LOOP 1720, the new iterated temperature for pressure level I is calculated and stored in the array T(I) via the equation T(I)=T(I)+FNUM/FDEN. In this equation, T(I) is the guess temperature at a given pressure level, while FNUM/FDEN is the temperature correction term. Note that this equation is exactly Equation 3 in Smith (1983).

After the temperature retrieval, if the first pressure level below the surface (IS) is greater than 850 mb, the lower levels of the retrieved temperature profile from 850 mb down to level IS are blended with the surface air temperature in DO LOOP 1780. Then, the temperature at all levels from IS down to 1000 mb (level 40) is set equal to TSTA in DO LOOP 1820, after which the retrieval mixing ratio, saturated mixing ratio and dewpoint for
each pressure level between 300 and 1000 mb inclusive is calculated and stored in the arrays W, S and TD, respectively, in DO LOOP 1860. Finally, after a jump to statement 1885 (remember, SRET is still on the FIRST iterative pass), a check is performed to see if the temperature profile satisfies the convergence criterion. The actual test occurs after DO LOOP 1920 (IF(SUM.LT.0.025) GO TO 1940).

If the profile satisfies the convergence criterion after the first temperature iteration, which is possible in unusual cases, the program skips to statement 1940. Otherwise, assuming the convergence criterion is not satisfied, SRET returns to statement 1390. Note that ITER has been incremented in the process and is now equal to 1. SRET is now ready to begin the second temperature iteration of the first iterative pass.

Within DO LOOP 1520 this second time, only radiances for the CO2 bands being used (subroutine VASRTE), as well as weighting functions and DELT values, are recomputed (ITER=1, IPASS=0, INIT=1). No transmittance recalculations (subroutine VASTAUR) are done. Then, since ITER .GT. 0, SRET jumps to statement 1580 and then eventually to 1670, after which a second temperature correction to the result of the first temperature iteration (DO LOOP 1720) is carried out. Following this DO LOOP, SRET proceeds like the first temperature iteration down through the rechecking of the convergence criterion. If the criterion fails a second time, control transfers once again to statement 1390, where the same steps as above are done again. The temperature iterations on the first iterative pass repeat until either the retrieved
temperature profile satisfies the convergence criterion, or the number of iterations (ITER) becomes greater than or equal to the iteration limit (ITLIM).

Assuming the convergence criterion is satisfied before ITER becomes .GE. ITLIM, SRET skips to statement 1940, where level NLS is redefined using the new retrieved temperature profile via DO LOOP 1942 (am assuming IENH .EQ. 0). This level will be used in the upcoming water vapor profile enhancement. Once NLS has been determined, transmittances for bands 7-10 are recalculated (VASTAU), a water vapor profile enhancement is performed via subroutine VWRET, and the total precipitable water vapor for the now enhanced retrieval mixing ratio profile is calculated. Finally, the dewpoints for levels 26-40 (300-1000 mb) are determined once again in DO LOOP 1940. Then, IPASS is set equal to 1, meaning one iterative pass has now been completed, and SRET once again returns to statement 1390.

At this point, to simplify the discussion which follows, I will assume that the retrieval is clear, that is, that IFLG has been set to 0. After following through the clear case, the user hopefully will be able to track through the cloudy (IFLG .NE. 0) case separately by him/herself.

During the SECOND iterative pass of SRET, assuming the original guess was NOT a climatology first guess, DO LOOP 1520 recomputes transmittances for bands 4 and 5 only. Then, the radiance, weighting function and DELT values are calculated again, and SRET skips to statement 1580 (ITER .GT. 0). At this point, within DO LOOP 1661, if the absolute value of the DELT
value for any CO2 band is .GT. 2.5, SRET jumps to statement 2200, and the retrieval is aborted. Assuming this problem does not occur, another convergence check is performed in DO LOOP 1665 (remember, IPASS=1 now) to see whether the absolute value of the difference between the observed and calculated brightness temperature for each CO2 band (DELT(I)) is .LE. to the expected noise of the brightness temperature observation for that band.

If all the bands pass this second convergence test, the retrieval process is done, and SRET jumps to statement 1951, where the post-retrieval process (calculating heights, stability indices, preparing the output array to be stored in the retrieval MD file, etc.) begins.

On the other hand, if any CO2 band fails the test, a temperature profile enhancement is attempted via subroutine VTRET (remember, IFLG=0). If the enhancement is successful (I0K set to +1 by VTRET), SRET jumps to statement 1730, where the lower portion of the temperature profile is again adjusted, and the mixing ratio, saturated mixing ratio and dewpoint arrays (W, S and TD) are all recalculated. IPASS is then incremented to 2, and the program returns once again to statement 1390. SRET is now on the THIRD iterative pass.

On the other hand, if the afore-mentioned temperature enhancement fails (I0K set to -1), SRET jumps to statement 1882, where IPASS is incremented to 2, and then jumps back to statement 1668, where a second temperature profile enhancement is attempted. Success on the second enhancement causes a jump to 1730 again and an eventual jump to 1390, while failure causes a
repeat jump to 1882.

The temperature enhancements will continue until either a successful enhancement is accomplished (and a corresponding jump taken to statement 1390), or the variable IPASS is incremented to 4 (indicating a third failed temperature enhancement).

At this point, SRET will be at one of two places. It will be at statement 1390, beginning either the third (IPASS=2), fourth (IPASS=3) or fifth (IPASS=4) iterative pass, if a successful temperature enhancement was accomplished BEFORE a third temperature enhancement failure. On the other hand, if there was a third consecutive temperature enhancement failure, SRET will be at statement 1882. These two locations in the SRET SUMMARY are, respectively, "ITERATIVE PASS #3, #4 or #5" (the letter "E"), or the letter "D," subheading 2b2b.

At this point, it is felt that any further discussion of SRET, in terms of the retrieval process, is too cumbersome. As a result, the user is invited to examine the SRET SUMMARY, together with SRET itself, to get a complete understanding of the remaining retrieval process. I, however, will now turn my attention to the code beginning at statement 1951, which is the initial statement in the post-retrieval portion of SRET.

Even at this late point, it is possible for the retrieval to fail. The retrieval will fail if the absolute DELT value for band KSAV (either band 5 or 6; see DO LOOP 1920 and code immediately preceeding DO LOOP 1560) is greater than the variable GTST. Assuming this check passes, subroutine HTX calculates heights from the retrieved temperature and moisture profiles. Then, after
the arrays PST, TST and TDST are filled, a call to subroutine
SNDANL calculates several different stability indices, such as
lifted index, total totals index, etc. Next, the heights computed
via subroutine HTX (the array S) are read into the array U, and
two thickness values (850-500 and 850-200 mb) are calculated and
stored in variables DELZ1 and DELZ2, respectively. Finally, an
averaged total-totals index is computed and stored in variable
TOTLS/ITOTLS, the variable NOGO is set to 0, meaning a successful
retrieval was performed, and SRET is ready to prepare an output
retrieval report array for insertion into the retrieval MD file.

The final stage of the post-retrieval processing consists
mainly of inserting the retrieval data into a buffer array IRETD
(note equivalence of to the array IRET near code line 50), which
is then inserted into column NN of the retrieval MD file via
subroutine VRTO. Some of the retrieval quantities placed in
IRETD include user mod flag (=0 now), total precipitable water
vapor, surface skin temperature, total totals index, relative
humidities calculated using bands 7, 9 and 10, and geopotential
heights. In addition, many of the retrieval temperatures and
dewpoints are also stored. Finally, surface values of pressure,
temperature, dewpoint, etc. are placed into IRETD to complete the
output array.

Lastly, NRET (number of reports) is incremented by 1 and, if
the plot option is in effect (IPLT=1), the total totals index
is plotted at the retrieval location in one of two colors, which
are usually blue-green or magenta, corresponding to clear
(KOLOR=2) or overcast (KOLOR=1, IFLG .NE. 0) retrievals,
respectively. After this, the final data is put into the retrieval MD file by VRTIO, and SRET returns to attempt the next retrieval.
SRET SUMMARY FROM STATEMENT 1390 TO STATEMENT 1951

ITERATIVE PASS #1 (IPASS=0)

A.) TEMPERATURE ITERATION #1: transmittance, lower level for radiative transfer, radiance, weighting functions (WF), DELT, band 6 test, cloudy channel test, temperature retrieval (T. ret.), convergence
   successful? go to 1940
   otherwise, go to 1390 and do TEMPERATURE ITERATION #2

B.) TEMPERATURE ITERATION #2: radiances, WF, DELT, T. ret., convergence
   successful? go to 1940
   otherwise, go to 1390 and do TEMPERATURE ITERATION #3

Temperature iterations are done like TEMPERATURE ITERATION #2 until convergence satisfied, or number of TEMPERATURE ITERATION becomes .GE. ITLIM (iteration limit). In either case, SRET will end up at statement 1940.

C.) 1940: first moisture enhancement
   total precipitable water vapor (TPWV) calculations
   set IPASS to 1 and go to 1390 for SECOND ITERATIVE PASS

ITERATIVE PASS #2 (IPASS=1) (assume retrieval is clear (IFLG=0) from this point on)

D.) transmittance for bands 4 and 5 only, radiance, WF, DELT, test DELT values (DO LOOP 1661) and quit retrieval if too poor; otherwise, test DELT values again (DO LOOP 1665)
1.) if all CO2 band DELT values small enough, retrieval processing done, and begin post-retrieval code at 1951
2.) otherwise, if failure in any CO2 band: first temperature profile enhancement

2a.) if temperature enhancement successful, go to 1730, adjust lower retrieval levels, get mixing ratios (W), saturated mixing ratios (S) and dewpoints (TD), set IPASS to 2 and go to 1390 to begin ITERATIVE PASS #3

2b.) if not successful, go to 1882, set IPASS to 2, and do second temperature enhancement

2bl.) if second temperature enhancement successful, do same as 2a above, but IPASS will = 3 when jump is taken to statement 1390; that is, SRET will be on ITERATIVE PASS #4

2b2.) if temperature enhancement fails, go to 1882, increment IPASS to 3, and do third temperature enhancement

2b2a.) if third temperature enhancement successful, do same as 2a above, but IPASS will = 4 when jump is taken to statement 1390; that is, SRET will be on ITERATIVE PASS #5

2b2b.) if temperature enhancement fails, go to 1882, increment IPASS to 4, set IOK to 0, raise ITLIM by 3, go to 1670, do T. ret., low level adjustments, get W, S and TD, check convergence criterion again
successful? go to 1940 and eventually end up at 1951
Temperature iterations done until convergence criterion satisfied, or number of TEMPERATURE ITERATION becomes .GE. ITLIM (the iteration limit). In either case, SRET eventually ends up at statement 1940.

1940: second moisture enhancement
TPWV calculations
jump to 1951 to begin post-retrieval code

ITERATIVE PASS #3, #4 or #5 (IPASS=2, 3 or 4)
E.) at statement 1390, no transmittance, WF, DELT, 1661 test
1.) if IPASS = 2 or 3, do 1665 test:
a.) if all bands converged, retrieval processing done, and begin post-retrieval processing at 1951
b.) otherwise, if a failure in any CO2 band, do another temperature enhancement
bl.) if temperature enhancement successful: low level adjustment, W, S and TD, increment IPASS
to 3 or 4

bla.) if IPASS=3, go to 1390, radiance, WF, DELT, 1661 test, 1665 band convergence test
bla1.) successful? go to 1951 and begin post-retrieval code
bla2.) unsuccessful? another temperature enhancement
bla2a.) if temperature enhancement successful, low level adjustment, W, S and TD, increment IPASS to 4, go to 1390, radiance, WF, DELT, 1661 test, jump to 1940 and then end up at 1951
bla2b.) if temperature enhancement unsuccessful, increment IPASS to 4, set IOK to 0, raise ITLIM by 3, go to 1670, T. ret., low level adjustments, W, S and TD, convergence criterion again
bla2bl.) successful? go to 1940 and end up at 1951
bla2b2.) unsuccessful? return to 1390, radiances, WF, DELT, 1661 test, T. ret., low level adjustments, W, S and TD, convergence criterion test

Temperature iterations are done until convergence criterion satisfied, or the ITERATION
NUMBER becomes .GE. ITLIM (iteration limit). In either case, SRET eventually ends up at statement 1940 and then 1951.

b1b.) if IPASS=4, go to 1390, radiance, WF, DELT, 1661 test, jump to 1940, and then end up eventually at 1951

b2.) if temperature enhancement unsuccessful, increment IPASS to 3 or 4

b2a.) if IPASS=3, another temperature enhancement

b2a1.) if temperature enhancement successful, low level adjustments, W, S and TD, increment IPASS to 4, go to 1390, radiance, WF, DELT, 1661 test, jump to 1940, and then end up eventually at 1951

b2a2.) if unsuccessful, increment IPASS to 4, IOK=0, ITLIM raised by 3, go to 1670, T. ret., low level adjustments, W, S and TD, convergence criterion again

b2a2a.) successful? go to 1940 and end up at 1951

b2a2b.) unsuccessful? same as b1a2b2 above

b2b.) if IPASS=4, set IOK=0, ITLIM raised by 3, go to 1670, T. ret., low level adjustments, W, S and TD, convergence criterion again
b2b1.) successful? go to 1940 and end up eventually at 1951
b2b2.) unsuccessful? same as bla2b2 above

2.) if IPASS=4, jump to 1940, and then end up eventually at 1951
\[ \text{CHECK} = \text{AMAX1}(\text{HUM}_1, \text{HUM}_2) \]

Set retrieval type to cloudy if this condition is true.

ABS(WIN) > 2.5 AND \text{HUM} < \text{HUM}_1 \rightarrow IFLG = 1

For high ground due, throw out shortwave channel.

Set aside one channel for testing success of retrieval.

\[ \text{ITCH} = 5 \]

IFLG \neq \Phi OR \text{IT} \leq 5 \rightarrow \text{ITCH} = 6

Declared clear retrieval type.

\[ \text{C} \rightarrow \Phi \rightarrow \text{K} \]

No surface climatology first guess.

\[ \text{GES} = 1 \] AND \[ \text{I} \neq \Phi \rightarrow \text{K} \]

Default-type retrieval determined as being clear.

Increasingly opaque; weighting functions peak higher for increasingly opaque channel.

\[ I < KC_1 \]
\[ I \leq KC_2 \]
\[ \text{I} \leq KC_1 + 1 \]
\[ JJ = KC_2 + 1 - I \]
\[ J = JJ \]
\[ J = 5 \]
\[ JJ = 6 \]

Reverse order because channel is more transparent.

\[ \text{DELT} \rightarrow \text{DELT}(J) \]
\[ \text{DELT} = 0.25 \]

IF this condition is true, the band is not cloud contaminated.

\[ 155^\circ \]

IF \Phi satisfies skip out, but check even if 6 passes.
CALL VTQ (LBVF)
CALL ENKDE ((GIB), LBVF, IT85, ID85, IT58, ID58, 
IT5S, KOD1, ID22)
CALL VTQ (LBVF)

STATEMENT 2196
NOC0 = 0 Indicates 
That Retrieval Was 
Successful

STATEMENT 2260

RETRIEVAL FAILURE
IRETD(3) = 7777

F

SUCCESSFUL RETRIEVAL
IRETD(3) = φ

U

V

W

K = K1
K = K + 1
K = K2
K = K3
K = K4

DO Loops 2245 and 2246 : 
Calculate Final Retrieval 
TBG for Bands K1→K3, 
As Well As Bands 1, 2, 3, 
And 4

CALL VASMAU (T(J), TOTO, VZEN, 
TAU, ISAT, K)

CALL VASATE (TAU(T), T, TS, CRAD, B, 
TBG, UDBB, K, IIA)

CALL VASMAU (T(J), TOTO, VZEN, 
TAU(J,J), ISAT, K)

CALL VASATE (TAU(J,J), T, TS, 
CRAD, B, TBG, UDBB, K, IIA)

CALL PRECU (P(J,J), T)

CALCULATE FINAL VALUE OF 
TOTAL PRECIPITABLE WATER VAPE

Above Changed From N 
To Conform With HTX

STORE ONLY HEIGHTS 
Below Surface
Subroutines used by SRET:

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CALDAY-I</td>
<td>53</td>
<td>MOGW</td>
<td>105</td>
<td>TEKPUT</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ENKODE-I</td>
<td>54</td>
<td>MOVW</td>
<td>106</td>
<td>LWNEWF</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>VRTIO-I</td>
<td>55</td>
<td>CLEANV</td>
<td>107</td>
<td>LWSV</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MDNAME-I</td>
<td>56</td>
<td>II</td>
<td>108</td>
<td>PAGE</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PLINKIV-I</td>
<td>57</td>
<td>STC</td>
<td>109</td>
<td>PENMOV</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>TOSET-I</td>
<td>58</td>
<td>DWRITE</td>
<td>110</td>
<td>BOX</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>GETDAY-I</td>
<td>59</td>
<td>TRMNL</td>
<td>111</td>
<td>PENBEG</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>GETFRM-I</td>
<td>60</td>
<td>STDATM</td>
<td>112</td>
<td>WALK</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>GESPRO-I</td>
<td>61</td>
<td>RAOBIN</td>
<td>113</td>
<td>SATPOS</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>WMIX-I</td>
<td>62</td>
<td>CSRAOB</td>
<td>114</td>
<td>JMBWTF</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>SDEST-I</td>
<td>63</td>
<td>MNRAOB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>PRECW-I</td>
<td>64</td>
<td>TQMES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>VASTAU-I</td>
<td>65</td>
<td>ITOC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>VTQ-I</td>
<td>66</td>
<td>GETNAV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>VWRET-I</td>
<td>67</td>
<td>EPOCH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>VASRTE-I</td>
<td>68</td>
<td>GETGAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>VTRET-I</td>
<td>69</td>
<td>VASNAV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>HTX-I</td>
<td>70</td>
<td>DDEST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>SNDANL-I</td>
<td>71</td>
<td>SCRAOB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>SATTV-I</td>
<td>72</td>
<td>CSRAOS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>VASDGC-I</td>
<td>73</td>
<td>CSRAOM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>TVSAT-I</td>
<td>74</td>
<td>CSRAOZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>NVINIT-I</td>
<td>75</td>
<td>CSRAOP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>SATEAR-I</td>
<td>76</td>
<td>CSRAOR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>ENCODX</td>
<td>77</td>
<td>CSRAOI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>CONTNT</td>
<td>78</td>
<td>INTPTW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>LTQ</td>
<td>79</td>
<td>EXTEMP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>ZECNOW</td>
<td>80</td>
<td>CLMGES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>MOVW</td>
<td>81</td>
<td>VASGES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>CLEANA</td>
<td>82</td>
<td>SURGES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>TQ</td>
<td>83</td>
<td>GETSFV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>PRLINX</td>
<td>84</td>
<td>IGNAME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>PRCL</td>
<td>85</td>
<td>PROFIX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>PROPEN</td>
<td>86</td>
<td>PREAV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>PRPRPR</td>
<td>87</td>
<td>PREATV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>LOCK</td>
<td>88</td>
<td>ULMR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>PRWR</td>
<td>89</td>
<td>CO2TV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>PRD</td>
<td>90</td>
<td>H2OTV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>UNLOCK</td>
<td>91</td>
<td>CONTAV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>PRCL</td>
<td>92</td>
<td>O3TAV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>POST</td>
<td>93</td>
<td>GAMTAV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>BLKA</td>
<td>94</td>
<td>INITPL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>ABOPT</td>
<td>95</td>
<td>QGDASH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>DOPEN</td>
<td>96</td>
<td>DSHOFF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>ENCODE</td>
<td>97</td>
<td>PLOT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>DREAD</td>
<td>98</td>
<td>DSHON</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>DCLOSE</td>
<td>99</td>
<td>ENPLT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>LWCLLOSE</td>
<td>100</td>
<td>ENPT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>MOVW</td>
<td>101</td>
<td>PENADD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>LWMP</td>
<td>102</td>
<td>PACK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>LWPO</td>
<td>103</td>
<td>SENOOUT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>EDEST</td>
<td>104</td>
<td>ATOE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
GPVA

GPVA is used to display VAS guess profiles of temperature, dewpoint and mixing ratio at either the cursor location (default), or a keyed-in latitude/longitude position. Three different types of guess profiles can be displayed: the guess profiles in either a keyed in MD file or the MD file defined under heading MDNG in the VASTEXT file (default), climatology guess profiles, or radiosonde profiles. Note that if the guess in the guess MD file (MDNG) is to be displayed, the user must run program GSVA before executing GPVA.

First, the type of guess to be displayed (CGES), as well as the surface analysis and debug options, are determined. In general, a user will probably want to use surface analyses when generating retrievals, so he/she should operate GPVA in the surface analysis mode to get an idea of what this data looks like at a particular location. This can be accomplished simply by keying in a 0 for the second positional parameter, if any positional parameters beyond the second are also going to be keyed in, or by keying in nothing, if only the first positional parameter is going to be used. In addition, variable ICLIM is set to 0, 1 or 2, depending on the guess profile chosen (0 for keyed in or VASTEXT file-defined MD file, 1 for climatology, 2 for radiosonde profile). This variable is vital later when accessing the actual guess information, because it tells subroutine GESPRO which guess data is to be read. Then, after variables VLAT and VLON (latitude/longitude) have been determined (= 0 if positional parameters in third and fourth positions have not been keyed in),
and assuming the sounder area has been set with program VPVA (LUC(81), NE. 0), the VASTEXT file is opened, read and closed. One of the items read from the VASTEXT file is the guess MD file number MDNG (IDOC(38)) (see reference to MDNG in above paragraph).

At this point, the rest of the discussion of GPVA, beyond the code where the MD file number is set, will be broken down into two sections. In the first section, it will be assumed that the latitude and longitude for the guess have not been keyed in, while the second section will discuss the case when these quantities have been keyed in.

Assuming there are no latitude/longitude keyins (VLAT, VLON = 0), the cursor line and element positions (IL, IE) are determined via subroutine TVSAT, and a call to subroutine VASDAT (with VDAT(1) = -1.) returns navigation data only, including the latitude and longitude positions of the cursor. Next, subroutine HRTOPO returns the surface elevation and characteristic (land = 1, ocean = 0), using a global 10 minute resolution topography. Finally, subroutine GESPRO gathers the rest of the guess data, taking into account the guess option desired. The data returned by GESPRO includes station (surface) pressure, 1000 mb height, and surface temperature and dewpoint. In addition, GESPRO accesses 15 temperature values from 1000-10 mb, 6 dewpoint values from 1000-300 mb, and 6 mixing ratio values, also from 1000-300 mb. After GESPRO has finished, subroutines SDEST and OUTINT print the results on the CRT.

On the other hand, if the latitude and longitude values have been keyed in, there is no need to call subroutine VASDAT for
navigation information, and GPVA jumps immediately to statement 3 and then subsequently executes subroutine HRTOPO. From this point to the end, this option proceeds identically to the previously-discussed no latitude/longitude keyin option.
ACQUIRE GUESS PROFILE

GELM IN PRINTING GUESS REPORT

Call gespro (iclum, nosfc, mdng)

Call sdest (' lat lon elev psfa 21000 tsfc dsfc', φ)

Call outint (kout)

Call sdest (' level press temp dewp wmix ', φ)

Print level, pressure, t, td and w (mixing ratio) for pressure levels 1004, 958, 999, 574, 400 and 300 mb.

I=1

I=I+1

I≤6

180

A

Call outint (kout)

Print only level, pressure and t for pressure levels 258, 299, 158, 104, 75, 50, 30, 20 and 10 mb.

I=7

I=I+1

I≤15

200

B

Call outint (kout)

END
Subroutines used by GPVA:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TSNIO-I</td>
</tr>
<tr>
<td>2</td>
<td>DOPEN-I</td>
</tr>
<tr>
<td>3</td>
<td>DREAD-I</td>
</tr>
<tr>
<td>4</td>
<td>DCLOSE-I</td>
</tr>
<tr>
<td>5</td>
<td>TVSAT-I</td>
</tr>
<tr>
<td>6</td>
<td>VASDAT-I</td>
</tr>
<tr>
<td>7</td>
<td>HRTOPO-I</td>
</tr>
<tr>
<td>8</td>
<td>GESPRO-I</td>
</tr>
<tr>
<td>9</td>
<td>SDEST-I</td>
</tr>
<tr>
<td>10</td>
<td>OUTINT-I</td>
</tr>
<tr>
<td>11</td>
<td>II</td>
</tr>
<tr>
<td>12</td>
<td>STC</td>
</tr>
<tr>
<td>13</td>
<td>ENCODE</td>
</tr>
<tr>
<td>14</td>
<td>ENCODX</td>
</tr>
<tr>
<td>15</td>
<td>ABORT</td>
</tr>
<tr>
<td>16</td>
<td>CONTNT</td>
</tr>
<tr>
<td>17</td>
<td>LTQ</td>
</tr>
<tr>
<td>18</td>
<td>ZECONV</td>
</tr>
<tr>
<td>19</td>
<td>MOVB</td>
</tr>
<tr>
<td>20</td>
<td>CLEANA</td>
</tr>
<tr>
<td>21</td>
<td>TQ</td>
</tr>
<tr>
<td>22</td>
<td>PRLNX</td>
</tr>
<tr>
<td>23</td>
<td>PRCLOS</td>
</tr>
<tr>
<td>24</td>
<td>PROPEN</td>
</tr>
<tr>
<td>25</td>
<td>PRPRPR</td>
</tr>
<tr>
<td>26</td>
<td>LOCK</td>
</tr>
<tr>
<td>27</td>
<td>PRWR</td>
</tr>
<tr>
<td>28</td>
<td>PRRD</td>
</tr>
<tr>
<td>29</td>
<td>UNLOCK</td>
</tr>
<tr>
<td>30</td>
<td>PRCL</td>
</tr>
<tr>
<td>31</td>
<td>POST</td>
</tr>
<tr>
<td>32</td>
<td>BLKA</td>
</tr>
<tr>
<td>33</td>
<td>LWCLOS</td>
</tr>
<tr>
<td>34</td>
<td>MOVWC</td>
</tr>
<tr>
<td>35</td>
<td>LWMOP</td>
</tr>
<tr>
<td>36</td>
<td>ITOC</td>
</tr>
<tr>
<td>37</td>
<td>GETFRM</td>
</tr>
<tr>
<td>38</td>
<td>TRMNL</td>
</tr>
<tr>
<td>39</td>
<td>TOMES</td>
</tr>
<tr>
<td>40</td>
<td>MOVCW</td>
</tr>
<tr>
<td>41</td>
<td>NVINIT</td>
</tr>
<tr>
<td>42</td>
<td>GETNAV</td>
</tr>
<tr>
<td>43</td>
<td>EPOCH</td>
</tr>
<tr>
<td>44</td>
<td>GETGAM</td>
</tr>
<tr>
<td>45</td>
<td>VASNAV</td>
</tr>
<tr>
<td>46</td>
<td>DDEST</td>
</tr>
<tr>
<td>47</td>
<td>CLEANW</td>
</tr>
<tr>
<td>48</td>
<td>EDEST</td>
</tr>
<tr>
<td>49</td>
<td>MOVW</td>
</tr>
<tr>
<td>50</td>
<td>READD</td>
</tr>
<tr>
<td>51</td>
<td>PLNKIV</td>
</tr>
<tr>
<td>52</td>
<td>OPNA</td>
</tr>
<tr>
<td>53</td>
<td>LWOPEN</td>
</tr>
<tr>
<td>54</td>
<td>LWGET</td>
</tr>
<tr>
<td>55</td>
<td>SATEAR</td>
</tr>
<tr>
<td>56</td>
<td>SATPOS</td>
</tr>
<tr>
<td>57</td>
<td>ANGGET</td>
</tr>
<tr>
<td>58</td>
<td>SOLARP</td>
</tr>
<tr>
<td>59</td>
<td>ANGLES</td>
</tr>
<tr>
<td>60</td>
<td>READOF</td>
</tr>
<tr>
<td>61</td>
<td>RBYTSX</td>
</tr>
<tr>
<td>62</td>
<td>ZEROS</td>
</tr>
<tr>
<td>63</td>
<td>DDEST</td>
</tr>
<tr>
<td>64</td>
<td>RDTTRK</td>
</tr>
<tr>
<td>65</td>
<td>MOVC</td>
</tr>
<tr>
<td>66</td>
<td>STDATM</td>
</tr>
<tr>
<td>67</td>
<td>RAOBIN</td>
</tr>
<tr>
<td>68</td>
<td>CSHAOB</td>
</tr>
<tr>
<td>69</td>
<td>MNRAOB</td>
</tr>
<tr>
<td>70</td>
<td>SGRAOB</td>
</tr>
<tr>
<td>71</td>
<td>CSHAOS</td>
</tr>
<tr>
<td>72</td>
<td>CSHAOM</td>
</tr>
<tr>
<td>73</td>
<td>CSHAQOZ</td>
</tr>
<tr>
<td>74</td>
<td>CSHAOP</td>
</tr>
<tr>
<td>75</td>
<td>CSHAOR</td>
</tr>
<tr>
<td>76</td>
<td>CSHAQI</td>
</tr>
<tr>
<td>77</td>
<td>WMIX</td>
</tr>
<tr>
<td>78</td>
<td>INTPTW</td>
</tr>
<tr>
<td>79</td>
<td>EXTEMP</td>
</tr>
<tr>
<td>80</td>
<td>CLEMGES</td>
</tr>
<tr>
<td>81</td>
<td>CALDAY</td>
</tr>
<tr>
<td>82</td>
<td>VASGES</td>
</tr>
<tr>
<td>83</td>
<td>SURGES</td>
</tr>
<tr>
<td>84</td>
<td>GETSFPV</td>
</tr>
<tr>
<td>85</td>
<td>IGNAM</td>
</tr>
<tr>
<td>86</td>
<td>PROFIX</td>
</tr>
<tr>
<td>87</td>
<td>LWPO</td>
</tr>
<tr>
<td>88</td>
<td>LWNEWF</td>
</tr>
<tr>
<td>89</td>
<td>LWSO</td>
</tr>
<tr>
<td>90</td>
<td>JMBWTF</td>
</tr>
<tr>
<td>91</td>
<td>VRTopo</td>
</tr>
</tbody>
</table>
ESVA

This program, which works only for retrieval surface MD files (schema RSVC), edits surface reports at the cursor location, and operates mainly in either of two modes. First, it can delete any combination of 1000 mb height (Z100), mean sea level temperature (TSL) or surface dewpoint depression (DD) values from the retrieval surface MD file. Otherwise, keyword data values for any or all three of these parameters can be substituted in. In addition, different combinations of the two modes (deletion, substitution) can also be keyed in. Besides these two main modes, there is also an option to add a report (where none existed previously) to the retrieval surface MD file. As part of this option, it is possible to calculate and store the TSL value corresponding to a keyed-in TSFC (surface temperature) value.

The first major step in ESVA is the calculation of the cursor line and element positions (IL,IE) via subroutine TVSAT. Next, the VASTEXT file is read, the navigation is initialized, and the MD file and row numbers are set to IDOC (36) and (37), respectively, which correspond to the retrieval surface MD file and row numbers listed in the VASTEXT file. This MD file is then opened for read/write, and the row header corresponding to row MDR is read via function MDGET (I0K=MDGET(MDNO, MDR, 0, IOUT)). Then, after a test has been performed to assure that data exists in this row of the retrieval surface MD file, the keys for the file are read in via function MDKEYS. This allows ESVA (in DO LOOP 11) to (among other things) locate the positions within the
array IOUT which contain the values for a given MD file report of Z100, TSL and DD. These are the positions (words) that will later be updated with new values, or have their old values deleted. Following DO LOOP 11, the values keyed in for Z100, TSL and/or DD are read separately using function IKWP, and the value of variable ITEM is determined. If a particular parameter is to be deleted from a given report, it should be keyed in with a "1" (TSL=1, for example). On the other hand, if some new value is to be substituted into the report in place of the old value, it should be keyed in *100 for TSL, TSFC and DD, but only as the value itself for Z100. Note that if "ESVA" alone is keyed in, the program will respond with "NO ACTION INDICATED AT THIS LOCATION" and terminate.

After frame information is accessed (GETFRM) and the plot package is started by subroutine INITPL, the cursor dimensions in latitude and longitude are determined. First, the NW corner of the cursor in satellite coordinates (line, element--TVSAT) and then earth coordinates (latitude, longitude--SATEAR) is calculated, after which ESVA repeats the process for the SE corner of the cursor.

The next step involves the retrieval surface MD file itself. A large DO LOOP (DO LOOP 70) reads MREC (= IOUT(3) in row header, which was read approximately 100 code lines previously) MD file entries, determines which ones should be edited, and then proceeds to do the actual data editing itself. A given report will be edited ONLY if it falls within the bounds of the cursor box. (Note that the cursor dimensions, if large enough, may
encompass more than one surface report.) For a given report which is contained by the cursor, the TV coordinates (raster, pixel) are obtained via subroutines SATEAR and SATTV. Following this, the correct words of the array IOUT are updated with the deletions or new values of Z100, TSL and(or) DD, with the array subsequently being put back into the retrieval surface MD file via function MDPUT (IOK=MDPVT(MDNO, MDR, M, IOUT)). In addition, variable ITEM is plotted on the video screen (via subroutine PLTDIG) at the report location within the cursor, after which the program returns to examine the next surface MD file report to see if it is also encompassed by the cursor box.

After DO LOOP 70 is exited, and assuming at least one report was located within the cursor box (NDAT .NE. 0), control transfers to statement 100, where an option to pack the retrieval surface MD file will be carried out if keyword parameter PACK was keyed in equal to 1. Normally, this keyword will not be used, and ESVA terminates. This concludes the discussion of how ESVA can be used for editing existing data within a cursor box. The following paragraph describes how ESVA operates when data is to be edited into an area where no prior data existed.

If ESVA is to be used in this mode, the cursor should be placed in the desired data void before the program is executed. Up to DO LOOP 70, ESVA functions exactly the same way in this "data addition" mode as it did when reports within the cursor were to be edited. However, DO LOOP 70 will not do anything this time, since all the existing retrieval surface MD file reports will be outside the cursor boundaries. Hence, variable NDAT will
equal 0 when DO LOOP 70 is exited, and control will transfer to
statement 85. At this point, if a value or values of Z100, TSL
and(or) DD was(were) keyed in and are to be added to the
retrieval surface MD file, the latitude and longitude of the
cursor (ALAT, ALON) are determined by subroutine SATEAR. Then, if
TSFC was keyed in, a corresponding mean sea level temperature
value is determined. In either case (TSFC keyed or not keyed in),
ESVA eventually reaches statement 300, and the new report is
placed in the retrieval surface MD file one column beyond the
last previous report (column IOUT(3) + 1). In addition, the
updated row header is written into the MD file via function
MDPUT, and variable ITEM is plotted on the video screen via
subroutine PLTDIG near the new report location. Finally, the
PACK option is again encountered, and ESVA terminates.
ESVA

DATA MGS / ZZ985891890 \NOAT/\n
CALL TVSAT (/FRM, IRAS, IPIC, ILE, IEO, IJS, IJD, JT)

SAVE LOCATION FOR POSSIBLE ADDITION

CALL TSWIO /111111, IDOC/

READ FASTEXT CONTEXT FILE

SET UP AREA FOR IMAGE SAKE

OPEN AND READ FASTEXT FILE

CALL DOPEN (/DLIT (MFILE), LUN, LEN)

CALL TVSAT (/FRM, 005, 335, LTOP, LELE, ISS, IO, IT)

CALL TSVSAT (/FRM, 095, 335, LTOP, LELE, ISS, IO, IT)

CALL DREAD (LUN, ITERM, IDOC)

CALL DCLOSE (LUN)

INITIALIZE NAVIGATION

CALL NVINIT (BETAIN, BETDOT, INAV, PTIME)

SET MFILE and ROW #s to FASTEXT "M095" and "M335" respectively

OPEN RETRIEVAL SURFACE MFILE for READ/WRITE

MDOPEN (MORD, 2) = \N

CALL SDEST ('CANNOT OPEN MFILE NO. \MORD\')

END

READ ROW HEADER RECORD

I0K = MDGET (MDNO, MDR, 0, ND)

CALL SDEST ('CANNOT LOCATE ROW NO. \I0K\', MDR)

END

READ TEST RECORD

TEST TO SEE IF DATA EXISTS IN RETRIEVAL SRC. MFILE
SET VALUE TO "MISS"
IF "1" WAS KEPT IN
(ESYA. Z100 = 1, FOR EXAMPLE)

DETERMINE VALUE OF "ITEM" WHICH WILL
THEN BE SUBSEQUENTLY
PLOTTED OVER ANY EDITED
REPORTS OR A GIVEN
ADDED REPORT

CALL SDST ('NO ACTION INDICATED AT THIS LOCATION, 0)
GET FRAME INFORMATION

CALL GETFRM(IFRM, MF)

INITIALIZE PLOT PACKAGE

CALL INITPL(IWRK, 0)

LUC(B) = 0

FOR VAS GET CURSOR DIMENSIONS IN LAT/LON

CALL TVSAT(IFRM, IRAS, IPIC, IL, IE, JS, JD, JT)

CALL SATEAR (PTIME, FUN, FELE, FLAT, FLOW, 1, INAV, BEHIN, BETOT, 0.)

ILAMIN = LAT * 100 - 2000

ILAMAX = ILAMIN + 4000

CONVERT TO +W-E CONVENTION

ILAMAX = IROUND (FLAT * 10000.)

ILON = -LON * 100 - 2000

ILONMAX = ILON + 4000

DO LOWER RIGHT CORNER NOW

CALL TVSAT(IFRM, IRAS, IPIC, IL, IE, US, JD, JT)

CALL SATEAR (PTIME, FUN, FELE, FLAT, FLOW, 1, INAV, BEHIN, BETOT, 0.)

CONVERT TO +W-E CONVENTION

ILAMIN = IROUND (FLAT * 10000.)

ILAMIN = IROUND (FLON * 10000.)

IF KBUG 0 F CALL OUTINT(KOUT)
DO LOOP 5 IF CHECK ALL RETRIEVAL SURFACE MD FILE REPORTS TO SEE WHICH (IF ANY) LIE WITHIN THE CURSOR DIMENSIONS

\[ IOK = MDGET( MONO, MOR, M, IOUT) \]

GET MD FILE REPORT

\[ KBUG = \emptyset \]

CALL OUTINT( IOUT)

DELETE EVERYONE WITHIN CURSOR

PASS BY REPORTS WHICH LIE OUTSIDE CURSOR DIMENSIONS

\[ IOK = \text{MDGET}( \text{MONO, MOR, M, IOUT}) \]

70 B

70 B

THIS SOUNCING MUST GO

LOCATE REPORT TO BE EDITED IN TERMS OF TV (VIDEO SCREEN) RASTER/PIXEL COORDINATES

\[ LUC(I) = \emptyset \]

LOCATE RASTER AND PIXEL OR SOUNCING

REVERT TO -N +E CONVENTION

CALL SATEAR( PTIME, FPN, FELE, FLAT, FLOM, Z, INAV, BETAIN, BETOOF, \emptyset )

69 \emptyset

CALL SATTY( PFM, IL, IE, IRAS, IPIC, US, JO, JT )

\[ KBUG = \emptyset \]

CALL OUTINT( IOUT)

SET APPROPRIATE WORDS (FOR MISS, TLS AND OR 00) OF REPORT ARRAY "IOUT" EITHER TO "MISS" OR NEW VALUE

\[ IOK = \text{MDPUT}( \text{MONO, MOR, M, IOUT}) \]

PUT UPDATED REPORT BACK INTO RETRIEVAL SURFACE MD FILE

\[ IOK = \emptyset \]

CALL SDEST( 'SRCH GIVES IM=', IM )

70 B
* ADD KEPT-INC VALUES OF I, HH, TSL AND
  (OR) DD, OR THE VALUE OF TSL CALCULATED
  FROM A KEPT-INC VALUE OF TSPC, TO THE
  RETRIEVAL SURFACE MD FILE

* FIND SFC, ELEVATION
  FOR USE IN
  DETERMINING
  TSL FROM A KEPT-INC
  TSPC VALUE

CALL MRTOPO
  (MLAT, MLON, IEI, ICH)

CALL SDEST
  ("TROUBLE WRITING
  DATA RECORD NO. M")
Subroutines used by ESVA:

1) TVSAT-I  53) EDEST
2) DOPEN-I  54) MOVW
3) DREAD-I  55) II
4) DCLOSE-I  56) DWRITE
5) NWINIT-I  57) SATPOS
6) TSNIO-I  58) QGDASH
7) SDEST-I  59) DSHOFF
8) OUTINT-I  60) PLOT
9) GETFRM-I  61) DSHON
10) INITPL-I  62) PUC
11) SATEAR-I  63) WD
12) SRCH-I  64) ENPT
13) SATTV-I  65) PENADD
14) PLTDIR-I  66) PACK
15) ENDPLT-I  67) SENOUT
16) ENKODE-I  68) ATOE
17) JSOX-I  69) TEKPUT
18) HRTPO-I  70) ISQX
19) TRMNL  71) TOKANL
20) TQMES  72) SQSLED
21) BLKA  73) LWPO
22) STC  74) LWNEWF
23) ITOC  75) LWSO
24) MOVB  76) PAGE
25) LTQ  77) PENVIV
26) CLEANA  78) BOX
27) TQ  79) PENOBEG
28) PRLINX  80) WALK
29) PRCLOS  81) JMBWTF
30) PROPEN
31) PPRPRP
32) LOCK
33) PRWR
34) PRRD
35) UNLOCK
36) PRCL
37) POST
38) ABORT
39) ENCODE
40) ENCODX
41) CONTNT
42) ZECNV
43) LWCLOS
44) MOVWC
45) LWMP
46) GETNAV
47) EPOCH
48) GETGAM
49) VASNAV
50) DDEST
51) MOVWC
52) CLEANW
VTPW

VTPW is used to retrieve values of total precipitable water vapor (TPWV) for individual fields of view (FOVs) within a retrieval area, and can also be keyed in to generate an image of the results. Note that the resolution of these retrievals will almost always be much greater than the usual temperature/moisture retrievals discussed in SRAD/SRET or VTPZ, because those retrievals were made using brightness temperature (TBB) measurements averaged over anywhere from 1 to 121 FOV retrieval boxes. In other words, VTPW will have from 1 to 121 individual retrievals PER RETRIEVAL BOX. Note also near code line 30 which VASTEXT file pointers must be set before the execution of the program. Most of these pointers would be set by the time a typical retrieval area was processed; however, if VTPW is used BEFORE a given retrieval area has been processed, any pointers not set must be set using program SPVA. In addition, the pointers MDNR, MDRR and NRET (retrieval MD file and row numbers, and number of retrievals performed so far, respectively) must be set to the proper values with SPVA before VTPW is run, irrespective of whether a retrieval area has been processed previously or not. The reason for this is that the retrieval MD file used to store the results of VTPW must be created using a different schema (VTWV) than is used for typical temperature/moisture retrieval MD files (VRET). In other words, these three pointers will all necessarily have different values than their predecessors from the previous retrieval area, while the rest of the VASTEXT file pointers may or may not be the same. Finally, it will be assumed
that the area of retrievals to be processed will have its NW and SE corners determined by the cursor position and a keyin via keyword parameter END, respectively. (Note that keyword parameter BEG must be keyed in as BEG 0 0 for this option to work properly. This will become more clear as the discussion proceeds. Also, the cursor should be positioned near the NW corner of the image before VTPW is executed.)

Initially, the number of the digital area to be used for storing the image of the TPWV results is determined via keyword parameter ARA (NARA=IKWP('ARA',1,0)). ARA should normally be included in the program keyin. However, if it is not keyed in (NARA=0), no image of TPWV will be created or stored in the digital area. Next, subroutine VRTIO reads both the VASTEXT file and the retrieval MD file row header, after which variables MDNG (upper air guess MD file), MDNR and MDRR are set to VASTEXT file values (through common block DOC and hence the array IDOC). Then, after variable LRO (the total number of retrievals as recorded in the VASTEXT file) is determined, the type of guess to be used for the retrievals (climatology or grid (default)), as well as the surface analysis option, are stored in variables CGES and NOSFC, respectively. Next, the size (in FOVs) of one side of a given retrieval box is determined (11 by default if no value is keyed in with keyword parameter BOX)) and stored in variable NBXS. If the sounder area data is large detector in nature, keyword parameter BOX should be keyed in equal to 6. This will allow VTPW to set the box size to 5 (the typical size of large detector retrieval boxes) and variable IFILL to 1 (as opposed to 0 for
small detector data). For small detector data, since the usual size of small detector retrieval boxes is 11*11 FOVs, keyword parameter BOX usually will not be keyed in.

Following the box size determination, the line and element retrieval box spacings in FOVs (INCR1, INCRE), and the end and initial line and element coordinates defining a retrieval area (LLINE, LELEM, ILINE, IELEM) are set. Note that these coordinates, if not keyed in via keyword parameters END and BEG, will take on VASTEXT file values by default. At this point, several other variables, such as the debug option (LBUG) and plot option (IPLT) are established, after which the cursor location in TV coordinates (raster-INRAS, pixel-INPIC) are determined from User Common via function LUC.

After the row header for retrieval MD file MDNR has been written (put) into the file via function MDPUT (IOK=MDPUT(MDNR,MDRR,0,IRET)), a jump is taken to statement 140 (remember, keyword parameter BEG was keyed in as BEG 0 0), where subroutine TVSAT determines the retrieval area NW corner satellite coordinates from the cursor position. Then, since LLINE*LELEM .NE. 0 (remember, keyword para. END supposedly was keyed in as END line# element#), VTPW will jump to statement 200.

As an aside, the proceeding code between and including statements 120 and 200 will now be discussed in terms of some other modes of operation of VTPW. First, if both the initial and ending line and element coordinates have been keyed in via keyword parameters BEG and END, and the processing is being done on a video terminal (LUC(16)=1), a call to subroutine GETFRM will
return image frame information, and VTPW will jump to statements 160 and then 200 to process a retrieval area. On the other hand, if both the initial and final line and element coordinates have been keyed in, but the processing is being done on a non-video terminal (LUC(16)=0), the call to GETFRM will not occur before the area is processed. (This makes sense, because it is impossible to have image frame information for a non-video terminal.) Finally, a single retrieval box at the cursor location will be processed if keyword parameters BEG and END are both entered with two zeros.

Returning now to the previous discussion, VTPW calls subroutine VASDAT, which returns navigation data only, because VDAT(l) is set to -1 before VASDAT is called. Important information returned by VASDAT includes the line and element resolution of the sounder area image (ILRES,IERES), which are then used, in turn, to calculate the line and element spacing of each retrieval box (INCIL,INCIE). At this point, it would be a good idea to view the layout of a typical retrieval area by glancing at Diagram 3 immediately following this discussion.

Next, assuming the area number (NARA) is .NE. 0, the exact line and element coordinates of the NW and SE corners of the image (as opposed to ILINE/IELEM, LLINE/LELEM, which are the satellite coordinates for the CENTERS of the NW and SE corner retrieval boxes within the image) are calculated. Following this, the dimensions of the retrieval area, in terms of lines and elements, are calculated and placed in the variables MLIN and MELE, respectively. Then, a digital area and its attendant
directory are created via subroutines ARASIZ and ENAREA, after which the digital area itself is opened via subroutine OPNA. This area will be used to store the TFWV values retrieved by VTPW as an image.

After variables KLINES and KELEMS (number of FOVs from top to bottom and side to side, respectively, in a retrieval box) are set, VTPW enters DO LOOP 820, which comprises almost all of the remaining program. Nested within DO LOOP 820 are DO LOOPS 740, 640 and 620. Together, these four loops create the image of TPFW from the vapor totals retrieved for each FOV in the retrieval area. VAS data is accessed retrieval line by retrieval line within the retrieval area, retrieval box by retrieval box within each retrieval line, and from the NW to the SE corner within each retrieval box, moving successively FOV by FOV to the right within each scan line of a given box. (Again, see Diagram 3 at the end of this discussion.)

The VAS data itself is accessed by a call to subroutine VASDAT about 10 lines inside of DO LOOP 620 (CALL VASDAT(ILINES,IELES,VDAT)). After the call to VASDAT, several tests are performed to see if a retrieval should be attempted. Note that a retrieval will NOT be attempted if: latitude is .GE. 90 degrees, satellite zenith angle is .GT. 60 degrees, land elevation cannot be returned due to a problem with the land elevation file, or TBB data for that particular FOV is missing. DO LOOP 320 then checks bands 1 through 12, setting a given band to VMISG if its TBB value is unreasonable. If there is reasonable data, and variable IMF=0, variable MSAM is incremented by 1. MSAM
will be incremented only once per FOV, since IMF is set to 0 after the initial incrementation. In addition, assuming that neither the band 7 or 8 TBB data for the given FOV is missing, the spin budget for each band is saved in the array NSPIN.

Upon exiting DO LOOP 320, if MSAM has not been incremented (i.e., each band from 1 through 12 either has bad data or is missing), VTPW will move to statement 600, where the element of the array IARRAY (used for generating the image of TPWV) corresponding to the FOV in question is filled with the default TPWV image value of 255. After this, VTPW moves to the next FOV to attempt a new retrieval. For the first FOV to be processed in a given retrieval box (which will be the FOV in the center of the box), variable IGFLG=1, and subroutine GESPRO accesses the guess information, depending on the type of guess chosen (IGES). Within GESPRO, in addition to the accessing of the guess profiles of temperature and mixing ratio, several surface parameters are determined, including surface pressure, temperature, and dewpoint. It should be noted that only ONE set of guess profiles is accessed for each retrieval box, since variable IGFLG is set to 0 after the guess profiles are accessed successfully for the first time.

Assuming the guess for a given retrieval box has been accessed successfully (TGES(1).GT.0) the first time and the surface parameters have been determined, DO LOOP 360 calculates the first pressure level below the surface for the retrieval box and stores the result in the variables IS and NLS. Following this, DO LOOP 440 finalizes the guess temperature and moisture
profiles. The resulting guess temperature profile will be identical to the original guess above 300 mb, but will consist of the surface value (TSTA) blended with the original guess from 300 mb down to level NLS, and values equal to the surface value itself from the first level below NLS down to 1000 mb. In addition, the resulting guess moisture profile will be equivalent to .02 g/kg above 100 mb, but will equal (.2)*(saturated value) from 100-250 mb, a blend between the surface value (WSTA) and the original guess from 300 mb down to level NLS, and the surface value itself again from the first level below NLS down to 1000 mb.

After the guess profile (if LBUG .NE. 0) has been printed on the line printer, and other information concerning the FOV location and zenith angles has been displayed (depending on the value(s) of variables LCRT and LBUG) on either the CRT or line printer, subroutine VASTPW calculates the TPWV amount for the FOV in question, returning the result via output parameter URET. The maximum possible value of URET allowed by VASTPW is 12 cm.

Assuming that VASTPW is successful, variable NDONE is incremented by 1. Note that only every "nth" retrieval will be outputted to the retrieval MD file (i.e., when variable MDOUT=1). After statement 560, assuming the plot option is in effect (IPLT.NE.0), subroutine SATTV calculates the FOV raster/pixel coordinates, after which subroutine WASDVG plots the TPWV value on the video screen at that location. Then, assuming the report is to be written into the retrieval MD file (MDRR.GE.0, MDOUT.EQ.1), several elements of the output array IRET are filled
with retrieval data, after which the entire array is put into column LRO of the retrieval MD file via function MDPUT. Following this, the retrieval MD file row header is updated with the new number of completed TPWV retrievals via a second MDPUT. Finally, DO LOOP 620 concludes by updating several arrays with TBB, TPWV and fail (IFAIL) flag data. Note in particular the data stored in the array IARRAY. This data will be used later to generate the image of TPWV. (As an aside, an image value of 0 is pure black, while a value of 255 is pure white.)

Following DO LOOP 640 (entire retrieval box completed at this point), and assuming the VAS data is small detector (IFILL.EQ.0), VTPW enters DO LOOP 700. This loop, together with DO LOOP 680, is designed to fill in blank scan lines within a given retrieval box column by column with an estimated TPWV image value (remember the relative widths of scan lines and retrieval lines). The estimated value is based on TPWV information from bordering scan lines, as well as band 8 TBB data (which exists for ALL of the FOVs in the retrieval area, regardless of whether the data is small or large detector).

Finally, DO LOOPs 780 and 760 prepare the TPWV image results stored previously in the array IARRAY for input into digital area NARA. DO LOOP 760 prepares the array IARET for input into subroutine PACK. After DO LOOP 760 has been completed, IARET will contain TPWV image information for all the FOVs in one scan line of the retrieval area. Then, subroutines PACK and WRITA together write the scan line of image data into digital area NARA. After DO LOOP 780 has been completed, a message saying FINISHED LINE is
displayed on the CRT, and VTPW moves to the next retrieval line within the retrieval area (DO LOOP 820 increments).

After all the retrieval lines in the retrieval area have been processed, the VASTEXT file is updated with the total number of retrievals generated (stored in variable LRO). This is accomplished by the call to subroutine VRTIO just before statement 840.

Finally, the output digital area (NARA) is closed, and a summary of the failures encountered during the processing of the retrieval area (the array NFAIL) is printed on the CRT. At this point, after VTPW has finished, system program DF can be used to load digital area NARA into a given image frame (i.e., it can be used to display the previously-generated image of TPWV on the video screen).
DIAGRAM 3

TYPICAL RETRIEVAL AREAS

---

**small detector**

SATellite Coordinate
ELEMENTS (INCREASING)

\[ \rightarrow \]

SATellite Coordinate
LINES (INCREASING)

\[ \downarrow \]

RETRIEVAL
LINE SPACING (INCIL)

\[ \times \]

INCIL

\[ \downarrow \]

INCIL

\[ \times \]

- INCIL \rightarrow (RETRIEVAL BOX SPACING)
- NEXT RETRIEVAL LINE

\[ \rightarrow \]

- INCIL \rightarrow

\[ \times \]

- INCIL \rightarrow

NOTES:
1) DIMENSION OF RETRIEVAL BOX = 11 x 11 FOVS
2) THIS RETRIEVAL AREA HAS 4 RETRIEVAL BOXES PER RETRIEVAL LINE, WITH EACH RETRIEVAL BOX CONTAINING 121 FOVS. FOR THE FIRST RETRIEVAL LINE, THERE ARE 77 FOVS WHICH HAVE SMALL DETECTOR DATA (4x11 x 3x111), MEANING 44 FOVS HAVE NO SMALL DETECTOR DATA. FOR THE SECOND RETRIEVAL LINE, THE NUMBERS CHANGE TO 55 AND 66, RESPECTIVELY. IT IS POSSIBLE TO HAVE ANYWHERE FROM 4 TO 7 SCAN LINES OF VALID DATA WITHIN A GIVEN 11 x 11 RETRIEVAL BOX.

---

**large detector**

LINES

\[ \downarrow \]

INCIL

\[ \times \]

- INCIL \rightarrow

\[ \downarrow \]

- INCIL \rightarrow

1 Scan Line of Data

1 Scan Line of Data

NOTES:
1) DIMENSION OF RETRIEVAL BOX = 5 x 5 FOVS
2) THIS RETRIEVAL AREA HAS 4 RETRIEVAL BOXES PER RETRIEVAL LINE, WITH EACH RETRIEVAL BOX CONTAINING 25 FOVS, ALL OF WHICH CONTAIN VAS TBB DATA.
IGES = 1 FOR NO "IGES" KEVIN OR "GSS=G" KEVIN (Both options result in G610 first guess being used)

IFILL = Ø SMALL DETECTOR

Determine SURFACE OPTION
NOSFC = IKWP('SFC',1,IDE)
NBXS = IKWP('BOX',1,IDE)

Determine RETRIEVAL BOX SIZE
MOD (NBXS,2) = Ø LARGE DETECTOR
IFILL = 1

Set NUMBER OF FOV BETWEEN RETRIEVAL BOXES IN BOTH Y (INCAL) AND X (INCAE) DIRECTIONS
LLINE = IKWP('END',1,IDO)(5)
LELEM = IKWP('END',2,IDO)(5)

Set RETRIEVAL AREA LINE/ELEMENT BOUNDS
ILINE = IKWP('BGE',1,IDO(5))
IELEM = IKWP('BGE',2,IDO(5))

LBUG = IKWP('BUL';1,Ø)
LCAT = IKWP('CAT',1,Ø)

Options to PRINT INFORMATION ON LINE PRINTER (LBUG) OR CRT (LCAT)

PLOT OPTION
IPLT = IKWP('PLT',1,Ø)

Get CURSOR MASTER/PIXEL COORDINATES

Set RETRIEVAL MD FILE ROW HEADER
IOK = MDPUT(MDDR,MORR,20)

IOK # Ø CALL EDIST ('TROUBLE WRITING HEADER FOR ROW',MDRR)

CALL ABORT(9)

END
Subroutines used by VTPW:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CALDAY-I</td>
</tr>
<tr>
<td>2</td>
<td>ENKODER-I</td>
</tr>
<tr>
<td>3</td>
<td>VRTICI-I</td>
</tr>
<tr>
<td>4</td>
<td>GETFDAM-I</td>
</tr>
<tr>
<td>5</td>
<td>ARASISZ-I</td>
</tr>
<tr>
<td>6</td>
<td>OPNA-I</td>
</tr>
<tr>
<td>7</td>
<td>VTQ-I</td>
</tr>
<tr>
<td>8</td>
<td>GESPRO-I</td>
</tr>
<tr>
<td>9</td>
<td>SDST-I</td>
</tr>
<tr>
<td>10</td>
<td>VASDAT-I</td>
</tr>
<tr>
<td>11</td>
<td>ENAREA-I</td>
</tr>
<tr>
<td>12</td>
<td>LTO-I</td>
</tr>
<tr>
<td>13</td>
<td>VASTPW-I</td>
</tr>
<tr>
<td>14</td>
<td>SATTO-I</td>
</tr>
<tr>
<td>15</td>
<td>VASDIG-I</td>
</tr>
<tr>
<td>16</td>
<td>PACK-I</td>
</tr>
<tr>
<td>17</td>
<td>CLOSAO-I</td>
</tr>
<tr>
<td>18</td>
<td>EDEST-I</td>
</tr>
<tr>
<td>19</td>
<td>TVDAT-I</td>
</tr>
<tr>
<td>20</td>
<td>WRITA-I</td>
</tr>
<tr>
<td>21</td>
<td>ABORT-I</td>
</tr>
<tr>
<td>22</td>
<td>ENCODX</td>
</tr>
<tr>
<td>23</td>
<td>CONTNT</td>
</tr>
<tr>
<td>24</td>
<td>LTQ</td>
</tr>
<tr>
<td>25</td>
<td>ZECONV</td>
</tr>
<tr>
<td>26</td>
<td>MOV</td>
</tr>
<tr>
<td>27</td>
<td>CLEANA</td>
</tr>
<tr>
<td>28</td>
<td>TQ</td>
</tr>
<tr>
<td>29</td>
<td>PRLNX</td>
</tr>
<tr>
<td>30</td>
<td>PRCLX</td>
</tr>
<tr>
<td>31</td>
<td>LOCK</td>
</tr>
<tr>
<td>32</td>
<td>PRWR</td>
</tr>
<tr>
<td>33</td>
<td>PRRD</td>
</tr>
<tr>
<td>34</td>
<td>UNLKC</td>
</tr>
<tr>
<td>35</td>
<td>PRCL</td>
</tr>
<tr>
<td>36</td>
<td>POST</td>
</tr>
<tr>
<td>37</td>
<td>PROPE</td>
</tr>
<tr>
<td>38</td>
<td>BLKA</td>
</tr>
<tr>
<td>39</td>
<td>PRPRA</td>
</tr>
<tr>
<td>40</td>
<td>DOPEN</td>
</tr>
<tr>
<td>41</td>
<td>DREAD</td>
</tr>
<tr>
<td>42</td>
<td>MOVCW</td>
</tr>
<tr>
<td>43</td>
<td>II</td>
</tr>
<tr>
<td>44</td>
<td>STC</td>
</tr>
<tr>
<td>45</td>
<td>DWRITE</td>
</tr>
<tr>
<td>46</td>
<td>ENCODX</td>
</tr>
<tr>
<td>47</td>
<td>DCLSOE</td>
</tr>
<tr>
<td>48</td>
<td>LWCLSO</td>
</tr>
<tr>
<td>49</td>
<td>MOVW</td>
</tr>
<tr>
<td>50</td>
<td>LWNOP</td>
</tr>
<tr>
<td>51</td>
<td>LWPO</td>
</tr>
<tr>
<td>52</td>
<td>TRNML</td>
</tr>
<tr>
<td>105</td>
<td>MOVC</td>
</tr>
<tr>
<td>107</td>
<td>VASTAU</td>
</tr>
<tr>
<td>109</td>
<td>PREATV</td>
</tr>
<tr>
<td>111</td>
<td>CO2TAV</td>
</tr>
<tr>
<td>113</td>
<td>CONTAV</td>
</tr>
<tr>
<td>115</td>
<td>GAMTAV</td>
</tr>
<tr>
<td>117</td>
<td>INITPL</td>
</tr>
<tr>
<td>119</td>
<td>QGDASH</td>
</tr>
<tr>
<td>121</td>
<td>PLOT</td>
</tr>
<tr>
<td>123</td>
<td>WALK</td>
</tr>
<tr>
<td>125</td>
<td>PENBEG</td>
</tr>
<tr>
<td>127</td>
<td>ENPT</td>
</tr>
<tr>
<td>129</td>
<td>ATOE</td>
</tr>
<tr>
<td>131</td>
<td>BOX</td>
</tr>
<tr>
<td>133</td>
<td>ENDPLT</td>
</tr>
<tr>
<td>135</td>
<td>CLOA</td>
</tr>
<tr>
<td>137</td>
<td>JMBWTF</td>
</tr>
</tbody>
</table>
Acknowledgements

I wish to thank Dr. W. P. Menzel of the NOAA/NESDIS Advanced Satellite Products Project for his guidance and assistance throughout this project. In addition, Dr. C. M. Hayden of the NOAA/NESDIS Development Laboratory was indispensable in helping me understand the software itself, as were Dr. W. L. Smith and Messrs. H. M. Woolf and F. W. Nagle, of the NOAA/NESDIS Development Laboratory. Mr. G. S. Wade of the NOAA/NESDIS Development Laboratory helped extensively with Chapter 1 in addition to answering questions from time to time. Mr. P. Raphael helped with the flowcharting, and Messrs. A. J. Schreiner and T. H. Achtor also provided timely assistance.

Appendix I

General McIDAS Terminology

Appendix I contains descriptions of many of the McIDAS terms used in the manual, and consists of two sections. Section I consists of brief descriptions of many terms the user is likely to encounter while processing VAS retrievals. If the user has access to a McIDAS User's Manual, he/she may wish to augment Section I with the Manual's glossary. Section II, on the other hand, deals with file structure, and details the structure of files used during the retrieval process, including the VASTEXT file, User Common, MD files, grid files/grids, sounder areas, and digital areas.

I.) McIDAS Terminology

1.) alphanumeric display monitor (CRT): the display device used to present McIDAS programs, program outputs, etc., as opposed to images and (or) graphics (see also COLOR-VIDEO DISPLAY MONITOR).

2.) area: a section of disk which contains digital information; is used to store the representation of an image, in which the data is arranged by location in a 2-dimensional array; there are several types of areas, including

   A.) VISSR (Visible Infrared Spin-Scan Radiometer) area: stores either a visible image or an 11-micron IR image
   B.) VAS area (sounder area): stores VAS radiance information in anywhere from 1-12 bands

(see also SOUNDER AREA, VISSR AREA, or DNGSWT and DNSOUND from Section II).

3.) background: refers to McIDAS jobs that are submitted from
editor  mode which are usually heavily involved with numerical computation and development; as a result, background programs usually take longer to run than foreground jobs (see also FOREGROUND).

4.) band: a very narrow range of wavelengths to which the VAS instrument is sensitive (for example, the 11 micron band refers to the measurement of radiation in a very narrow band centered at 11 microns).

5.) color-video display monitor (TV): the display device used to view VAS images, graphics, etc., as opposed to program text and (or) output (see also ALPHANUMERIC DISPLAY MONITOR (CRT)).

6.) column: refers to a column within a given type of MD file (surface, retrieval, etc.) within which a certain number of reports exist, and is one of the coordinates used to locate reports within MD files (other coordinate is the row) (see also ROW, as well as DNMDDOC in Section II).

7.) column header: describes location (e.g., station number, latitude/longitude, elevation, etc.) for all reports at a particular geographic location (all reports in given column are from same geographic location), as well as number of reports in that column (see also DNMDDOC in Section II).

8.) coordinate system: a particular system of planar coordinates (x,y) for keeping track of location of surface reports, retrievals, etc.; there are 3 systems in use for the processing of VAS retrievals: satellite (line, element), TV (television) (raster, pixel) and earth (latitude, longitude); vertical coordinate is pressure in all 3 cases.
9.) cursor: a visual indicator which can be moved about the color-video display monitor either manually via the joysticks or automatically via a McIDAS keyin.

10.) digital area: refers to same entity as an area (see AREA).

11.) earth coordinates: (x,y) coordinates of given location measured relative to the earth, known obviously as latitude (y) and longitude (x).

12.) editor mode: one of 2 McIDAS system modes (other is McIDAS mode), is the mode in which programs can be modified using different edit commands, as well as executed.

13.) element coordinate: one of 2 satellite coordinates, gives location in the east-west (x) direction relative to the satellite image.

14.) foreground: refers to McIDAS jobs that are submitted from the McIDAS mode and are usually of an interactive nature; foreground jobs generally run quicker than background jobs (see also BACKGROUND).

15.) graphics: various depictions of maps, derived fields, parameter values, etc. that can be overlayed over different images (see also GRAPHICS FRAME).

16.) graphics frame: a color-video display monitor (TV) section of memory which is capable of holding the graphics for 1 image frame (note that a color-video display monitor has half as many graphics frames as image frames).

17.) grid: see DNGRID and DNGRIDS in Section II.

18.) grid file: see DNGRID and DNGRIDS in Section II.

19.) image: the radiance "picture" sensed by the VAS for a given
band (are therefore band 1 images, band 2 images, etc.) (see also DNGSWT).

20.) image frame: a color-video display monitor (TV) section of memory which is capable of holding an image of a given VAS band (note that color-video display monitors have a variable number of image frames, depending on the amount of memory accorded to the monitor in question).

21.) image loading: the act of placing a given VAS band's image in a certain color-video display monitor (TV) image frame via McIDAS program "DF" (see also IMAGE FRAME).

22.) initiator: refers to a part of system software which allows jobs to execute, a connection between a job and the system, are 10 initiators on the McIDAS system.

23.) joysticks: two small stick-like controls which are used to manually move the cursor; one control is for short, detailed movements, while the other is for movements which are longer, but less detailed.

24.) keylist: the list of keys for a schema which corresponds to a particular type of MD file (see also KEYS, SCHEMA).

25.) keys: alphanumeric labels which are exclusive to a given type of MD file (retrieval, upper air, surface, etc.) which represent various quantities stored within a given report from that type of MD file; for example, one might have a surface MD file, with the keys for this MD file's reports consisting of TSFC (surface temperature), TDSFC (surface dewpoint), Z100 (1000 mb height), etc.; then, a program which needed to access surface temperature from different surface reports would run through the
keylist corresponding to the schema for that particular type of
MD file, stopping when the key for surface temperature had been
determined (see also DNMMDOC in Section II).

26.) keyword parameter: a small "word" (the keyword) or sequence
of letters used to input data to a given McIDAS program, such as
MD file number, grid file number, MD file row number, etc.; some
examples would be: SPVA MDNR=4185, ESVA TSL=29100; keyword
parameters are usually thought of as being used to input optional
data to McIDAS programs (i.e., data secondary in importance to
that inputted via positional parameters).

27.) Large Word (LW) array file: refers to a file that is stored
on disk (MD files and gridfiles are 2 examples of LW array
files).

28.) line coordinate: one of 2 satellite coordinates, gives
location in the north-south (y) direction relative to the
satellite image.

29.) macro: a type of program which consists mainly of execution
of other programs, is usually used to consolidate the execution
of a series of related programs.

30.) McIDAS mode: one of 2 McIDAS modes (other is editor mode),
is the mode in which programs can only be executed, not modified
or edited.

31.) MD file: see DNMMDOC in Section II.

32.) navigation: refers to images; relating satellite coordinates
(line/element) in images to earth coordinates
(latitude/longitude).

33.) non-video terminal: McIDAS terminals which consist solely of
a keyboard input device coupled with an alphanumeric display
monitor (see also VIDEO TERMINAL).

34.) page: a section of a Large Word (LW) array file which
consists of 1024 4-byte words.

35.) pixel: the color-video display monitor (TV) coordinate in
the x-direction.

36.) pointer: refers to a word in User Common which identifies
the current sounder area, gridfile, or MD file for the user's
terminal (see also USER COMMON).

37.) positional parameter: a given variable is referred to as
such when it is included in the program keyin independent of any
keyword (as opposed to keyword parameters, where the value of a
given variable is entered in the program keyin in the company of
a small keyword or sequence of letters); in McIDAS programs where
both positional and keyword parameters are involved, positional
parameters are usually thought of as being used to input data
vital to the programs, while positional parameters are usually
used to input optional data; some examples would be VPVA 4181
and PLVA 2 500.

38.) raster: the color-video display monitor (TV) coordinate in
the y-direction.

39.) retrieval: the final temperature, moisture or total
precipitable water vapor profile (value) derived from a first
guess modified using the VAS radiance measurements (see also
SOUNDING).

40.) retrieval area: a subset of a given sounder area within
which retrievals are to be generated (can be smaller than or
equal to sounder area in size) (see also SOUNDER AREA).

41.) retrieval line: an east-west line of retrieval boxes within a given retrieval area (see also RETRIEVAL BOX, RETRIEVAL AREA).

42.) row: refers to a row within a given type of MD file (surface, retrieval, etc.) within which a certain number of reports exist, and is one of the two coordinates used to locate reports within MD files (other coordinate is the column) (see also COLUMN, as well as DNMDDOC in Section II).

43.) row header: describes a date and time (or sensor number, date and time) which is valid for all reports in a given row of an MD file (note that all reports in a given row are valid for same time), as well as the number of reports in that row (see also DNMDDOC in Section II).

44.) satellite coordinates: (x,y) coordinates of given location measured relative to the VAS image, known as lines (y) and elements (x).

45.) scaling: relates to manner in which different variable quantities are stored in grid/MD files; for instance, whether given variable is stored as 1/100, 1/10, 100, etc. times its true value.

46.) scan line: an east-west line of individual VAS fields of view within a given sounder or retrieval area (note that a retrieval line will almost always have a latitudinal width of several scan lines) (see also RETRIEVAL LINE).

47.) schema: the way in which reports from a given type of MD file are structured, with each quantity within the report being represented by a corresponding key, such as TSFC for surface
temperature, TDSFC for surface dewpoint, etc. (see also KEYS, KEYLIST; see also DNMDDOC in Section II).

48.) sounder area/file: an area used to store VAS radiance information in anywhere from 1-12 VAS bands; the user can think of a sounder area as a multi-spectral VISSR area, with each sounding location (field of view) being represented by 1-12 VAS measurements (see also DNGSWT and DNSOUND in Section II).

49.) sounding: the VAS radiance (brightness temperature) measurements for a given field of view, which are then used to generate the temperature/moisture retrievals (see also RETRIEVAL).

50.) spin budget: refers to the number of spins the geostationary satellite completes while the VAS instrument is sensing in a given band in order that a sufficiently strong signal-to-noise ratio is achieved.

51.) terminal: a McIDAS user's station which can be used to communicate with the system via a keyboard input; there are both video and non-video terminals.

52.) track: one circular band of storage area on a disk.

53.) TV: see COLOR-VIDEO DISPLAY MONITOR.

54.) TV coordinates: (x,y) coordinates of given location measured relative to the color-video display monitor, known as rasters (y) and pixels (x).

55.) User Common: terminal-dependent directory of pointers, terminal/cursor control information, etc. (see also DNUC in Section II).

56.) VAS pointer: the word of User Common which refers to the
sounder area of VAS radiances the user wishes to process (see also POINTER).

57.) VASTEXT file: a file used to store many quantities vital to the user during his/her McIDAS activities (retrieval processing, in particular), including retrieval MD file number, surface MD file number, sounder area number, etc.; the contents of this file can be displayed at any time on the CRT via program LOVA (see also DNVASTXT in Section II).

58.) video terminal: a McIDAS user's station with a keyboard input to the system, an alphanumeric display monitor (CRT), and a color-video display monitor (TV) (see also NON-VIDEO TERMINAL).

59.) VISSR area: a type of area which stores either a visible image or an 11-micron IR image (see also AREA).

60.) VRET: the schema used when creating the retrieval MD file which stores retrievals of temperature/moisture (see also SCHEMA).

61.) water vapor and totals imagery: refers to individual images of total precipitable water vapor or total-totals index generated by various VAS retrieval programs (total precipitable water vapor—VTPZ or VTPW; total-totals index—VTPZ) (see also IMAGE).

62.) weighting function: a mathematical expression which evaluates quantitatively the contribution to radiation of a given wavelength and satellite zenith angle at any atmospheric pressure level, is unique for each wavelength, pressure level and satellite zenith angle.
II.) File Structure

List of Documents in Section II: DNGRID
DNGRIDS
DNCSWT
DNMDDOC
DNSOUND
DNUC
DNVASTXT
GRID MARE THE `MCIDAS DATA STRUCTURES WHICH REPRESENT GEOGRAPHICALLY DISTRIBUTED DATA OF LOW RESOLUTION OR VOLUME (AS COMPARED TO SOURCES OR AREAS). THEY ARE N X N MATRICES, LIMITED TO A TOTAL SIZE OF N X N = 10240 (ABOUT 4 TIMES GREATER THAN ON THE HARRIS `MCIDAS SYSTEM). EACH NUMBER IN THE GRID IS AN OBSERVATION OR A CALCULATED VALUE OF A METEOROLOGICAL QUANTITY, WHOSE GEOGRAPHICAL COORDINATES ARE INFERRED FROM THE MATRIX COORDINATES AND INFORMATION ABOUT THE PROJECTION THAT IS KEPT IN THE HEADER ASSOCIATED WITH THE GRID.

GRIDS ARE GROUPED INTO "GRID FILES", UP TO 159 GRIDS PER FILE. THERE IS NO NECESSARY RELATIONSHIP AMONG THE GRIDS IN A GRID FILE; THE FILES JUST PROVIDE A CONVENIENT WAY OF HANDLING LARGE NUMBERS OF GRIDS. `MCIDAS PROVIDES FOR UP TO 9999 GRID FILES.

GRID FILE FORMAT

GRID FILES, LIKE MOST OTHER `MCIDAS FILES, ARE DESIGNED TO BE ACCESSED BY THE LOW-DISK I/O SUBROUTINES. THAT IS, THEY ARE THOUGHT OF AS A LARGE CONTINUOUS ARRAY OF WORDS, WITH THE FIRST WORD NUMBERED 0. THE WORD NUMBERS OF THE VARIOUS COMPONENTS ARE AS FOLLOWS:

WORD
-
0-9 FILE IO• 0-7 ARE 32 CHARACTERS OF (ERCDIC) LABEL INFORMATION. 4 IS THE PROJECT NUMBER OF THE USER WHO CREATED THIS FILE. 0 IS THE FILE CREATION DATE (YYDDD).

---------
64-127 HEADER (64 WORDS) FOR GRID #1 (FORMAT BELOW)
128-191 HEADER FOR GRID #2

... 10175-10233 HEADER FOR GRID #15
---------
10240-20477 GRID #1
20480-30759 GRID #2

---------

THE FIRST WORD OF THE GRID HEADER FOR GRID N IS WORD # 64 X N. THE FIRST WORD OF GRID N (THE MATRIX ITSELF) IS WORD # 10240 X N. AN EMPTY GRID SLOT IS REPRESENTED BY A NEGATIVE VALUE IN THE FIRST WORD OF THE HEADER, WHICH OTHERWISE GIVES THE TOTAL SIZE OF THE GRID.
GRID FILE DOCUMENTATION

GRID FILES ON IBM/MCIDAS ARE LARGELY A CARRYOVER FROM THE SAME
FILE STRUCTURE ON HARRIS/CDAS. A GRID STORES OBSERVATIONS MADE
AT STATIONS ON A GEOPHYSICAL GRID. THE LATTICE SHOULD BE THOUGHT
OF AS A CHEESECAKE SUPERIMPOSED ON A PROJECTION OF THE EARTH WHICH
IS EITHER (IN THE CURRENT IMPLEMENTATION) 1.--PSUEDO-MERCATOR
(i.e., LAT/LO) OR 2.--CONFORMAL. THE Recorder
OBSERVATIONS (HENCEFORTH CALLED THE "GRIDDED VARIABLE") ARE ASSUMED
MADE AT SOME (POSSIBLY VARIABLE) HEIGHT WHICH IS DESCRIBED BY THE
"LEVEL" PARAMETER IN THE GRID HEADER.

GRIDS ARE TYPICALLY MADE BY SOME PROCESS OF INTERPOLATION FROM AN
"M" ("METEOROLOGICAL DATA") FILE CONTAINING THE RAW OBSERVATIONS.

A GRID HAS 2 COMPONENTS: A 64-WORD HEADER WHICH DESCRIBES THE GRID
DATA, AND SO FORTH, AND THE ACTUAL GRID, WHICH IS JUST AN M X N-WORD
MATRIX IN WHICH "X" CANNOT EXCEED 10240.

A GRID FILE CAN CONTAIN UP TO 159 GRIDS, NUMBERED 1...159.
THERE IS ACTUALLY SPACE FOR A GRID "O", BUT THIS SPACE IS TAKEN UP BY:
THE COLLECTION OF 64-WORD GRID HEADERS. THESE HEADERS ARE ALSO NUMBERED
0...159 (HEADER 0 CONTAINS AN ARBITRARY 8-WORD ID FOR THE WHOLE FILE).
THE FORMULAE FOR LOCATING GRID HEADERS AND MATRICES IN LW-TERMS
(THAT IS, TREATING THE FILE AS A GIANT ARRAY WHOSE FIRST WORD IS
NUMBERED 0) ARE AS FOLLOWS:

HEADER 1 BEGINS AT WORD # 64 X 1.
MATRX 1 BEGINS AT WORD # 10240 X 1.

THE VALUE 10240 WAS CHOSEN TO BE BIG ENOUGH FOR PRACTICAL APPLI-
CATIONS (IT ADMITS GRIDS OF 100 X 100, 4 TIMES LARGER THAN THE MAXIMUM
SIZE ON IBM/MCIDAS), AND ALSO TO BE A MULTIPLE OF THE 1024-WORD DISK PAGE
SIZE ON IBM/MCIDAS. GRID MATRICES CAN BE SMALLER THAN 10240; THE
ACTUAL SIZE IS RECORDED IN THE GRID HEADER, BUT BECAUSE THE MCIDAS
DISK I/O PACKAGEALLOCATES PAGES ONLY AS NEEDED, SMALL GRIDS RESULT
IN THE CONSUMPTION OF ONLY A SMALL AMOUNT OF DISK SPACE, EVEN THOUGH
10240 "VIRTUAL WORDS" ARE RESERVED FOR EACH GRID MATRIX.

GRID TYPES:
CURRENTLY 2 TYPES OF GRIDS ARE DEFINED (THE TYPE REFERS TO THE
GEOPHYSICAL PROJECTION): PSUEDO-MERCATOR (LAT/LO) GRIDS, AND
CONFORMAL GRIDS. THE CONSTANTS DESCRIBING THESE 2 PROJECTIONS
ARE GIVEN IN WORDS 33-39 OF THE GRID HEADER. LAT/LO GRIDS ARE 2-DIM
MATRICES WITH 1-BASED (FORTRAN-TYPE) SUBSCRIPTS, THEN THE NORTHWEST
CORNER (UPPER LEFT-HAND CORNER) CORRESPONDS TO THE (1,1) COORDINATE;
LATITUDE DECREASES AS THE FIRST COORDINATE INCREASES; LONGITUDE
INCREASES AS THE SECOND COORDINATE INCREASES, AND THE MATRIX IS STORED
INTERNALLY BY COLUMN. YOU SHOULD FOLLOW THESE CONVENTIONS WHEN
CREATING GRIDS IF THEY ARE TO BE PROPERLY HANDLED BY THE GRID UTILITIES.
IMAGE TAPES PRODUCED BY UW/SSEC
VERSION 3.0, 4/84

THIS DOCUMENT DESCRIBES THE TAPE FORMAT DEvised BY UW/SSEC
FOR DISTRIBUTION OF SATELLITE IMAGERY TO EXTERNAL USERS.
THIS FORMAT IS DESIGNED TO BE READERLY TRANSPORTABLE TO A WIDE
CLASS OF MACHINES, AND PARTICULARLY EASILY TO MACHINES WITH 32-BIT
WORDS.

THE FORMAT IS CHARACTERIZED BY FIXED-LENGTH TAPE BLOCKS
WHICH ARE SELF-IDENTIFYING; THAT IS, EACH BLOCK CONTAINS WORDS
SPECIFYING THE TYPE OF THAT BLOCK AND THE POSITION OF THE BLOCKS
DATA WITHIN THE WHOLE DATA SET. FIXED-LENGTH BLOCKS MAKE IT EASIER
TO COMBINE DATA SETS OF VARYING TYPES ON A SINGLE TAPE, WITHIN
THE LIMITS OF IBM JOB-CONTROL LANGUAGE; AND THE SELF-IDENTIFYING
NATURE OF THE BLOCKS FACILITATES RECOVERY OF DATA AFTER A
BAD PATCH ON THE TAPE, WHICH IS IMPORTANT WHEN THE TAPE FORMAT
IS VIEWED AS AN ARCHIVE MEDIUM.

SATELLITE IMAGES AT SSEC

SATELLITE IMAGES PROCESSED AT SSEC OCCUR IN SEVERAL
DIFFERENT FORMS, DEPENDING ON THE DATA SOURCE. TWO OF THE MOST
IMPORTANT SOURCES ARE THE GOES VISSR AND GOES VAS INSTRUMENTS.
THE IMAGES FROM THESE INSTRUMENTS ARE REFERRED TO AND DOCUMENTED
IN DETAIL BELOW. REGARDLESS OF THE SOURCE, AN IMAGE CAN BE
CONCEIVED OF AS A SEQUENCE OF IMAGE "LINES" ARRANGED ONE BELOW
THE OTHER AND NUMBERED FROM TOP TO BOTTOM. THE TOP LINE BEING
NUMBERED ONE. EACH LINE CONSISTS OF A SEQUENCE OF IMAGE "ELEMENTS".
ARRANGED ACROSS THE LINE AND NUMBERED LEFT TO RIGHT. THE LEFT-
MOST ELEMENT BEING NUMBERED ONE. AN ELEMENT MAY BE A SINGLE
UNSIGNED 8-BIT QUANTITY, OR A SEQUENCE OF SIGNED 2'S COMPLEMENT 16-BIT
QUANTITIES, DEPENDING ON THE IMAGE FORMAT. (IN VAS IMAGES, THE
16-BIT QUANTITIES REPRESENT RAW OBSERVATIONS IN DIFFERENT SPECTRAL
BANDS AT THE SAME SPATIAL LOCATION.) IN ANY CASE, THE LINE/ELEMENT
NUMBERING SCHEME ASSIGNS A PAIR OF COORDINATES TO EACH
ELEMENT, CALLED THE "IMAGE COORDINATES" OF THE ELEMENT. THIS
COORDINATE SYSTEM IS DEFINED BY THE SATELLITE/CAMERA COMBINATION
AND IS INDEPENDENT OF HOW THE DATA MAY BE STORED ON DISK
OR TAPE.

AS AN EXAMPLE, GOES VISSR IMAGES IN THE VISIBLE-LIGHT BAND
CONTAIN 14568 LINES, EACH WITH 15288 ELEMENTS.

WHEN A SATELLITE IMAGE IS TO BE TRANSFERRED TO TAPE BY THE
MCIDAS SYSTEM, IT IS FIRST PREPARED IN A SPECIAL DISK STORAGE
REGION CALLED A "DIGITAL AREA", OR WHEN NO CONFUSION OF TERMS
WOULD RESULT, JUST AN "AREA".

ASSOCIATED WITH THE SYSTEM OF DIGITAL AREAS IS AN ADDI-
TIONAL DATA SET CALLED THE "AREA DIRECTORY". THE AREA DIRECTORY
ENTRY FOR A GIVEN AREA CONTAINS AUXILIARY INFORMATION DESCRIB-
ING IMPORTANT CHARACTERISTICS AND PARAMETERS OF THE AREA DATA,
INCLUDING THE IMAGE FORMAT. THIS AREA DIRECTORY ENTRY FOR
AN AREA IS ALSO INCLUDED ON THE TAPE AND IS DOCUMENTED IN DETAIL BELOW.

THE DATA STORED IN A DIGITAL AREA IS ARRANGED IN A TWO-DIMENSIONAL ARRAY, LIKE THE IMAGE WHICH IT REPRESENTS. EACH ELEMENT IN AN AREA HAS A LINE NUMBER, STARTING FROM ONE AT THE TOP, AND A ELEMENT NUMBER, STARTING FROM ONE AT THE LEFT. THE LINE/ELEMENT NUMBER PAIR DEFINES A COORDINATE SYSTEM FOR THE ELEMENTS IN AN AREA, CALLED "AREA COORDINATES".

IF EVERY ELEMENT IN EVERY LINE OF A SATELLITE IMAGE WERE ALWAYS STORED IN A DIGITAL AREA, THERE WOULD BE NO POINT IN DISTINGUISHING BETWEEN IMAGE AND AREA COORDINATES. THIS IS NOT USUALLY THE CASE HOWEVER, SINCE IMAGES ARE ORDINARILY TOO LARGE TO PROCESS EFFICIENTLY IN TOTAL. (THE VISIBLE-BAND IMAGE MENTIONED ABOVE WOULD REQUIRE OVER 200 MEGABYTES.) WHAT IS STORED IN AN AREA IS A RECTANGULAR SUBSET OF THE IMAGE, PRODUCED FROM THE ORIGINAL BY SAMPLING (IN THE VERTICAL DIRECTION) AND AVERAGING (IN THE HORIZONTAL DIRECTION). IN THE CASE OF VAS IMAGES, IT IS ALSO POSSIBLE TO ONLY INCLUDE IN THE DISK AREA A SUBSET OF THE AVAILABLE SPECTRAL BANDS, SO THAT EACH ELEMENT CONTAINS FEWER VALUES THAN ARE REPRESENTED IN THE ORIGINAL IMAGE.

THE AREA COORDINATES FOR AN ELEMENT ARE OBTAINED FROM ITS AREA COORDINATES BY THE FOLLOWING FORMULAS:

\[
\text{IMAGE LINE} = \text{ULTELE} + (\text{AREA LINE} - 1) \times \text{LINES}
\]

\[
\text{IMAGE ELEMENT} = \text{ULTELE} + (\text{AREA ELEMENT} - 1) \times \text{ELTELE}
\]

WHERE

\[
\text{ULTELE} := \text{IMAGE LINE} \times \text{AREA ELEMENT} (1,1)
\]

\[
\text{LINES} := \text{IMAGE ELEMENT} \times \text{AREA ELEMENT} (1,1)
\]

\[
\text{LINES} := \text{SPACING IN IMAGE LINES BETWEEN CONSECUTIVE AREA LINES}
\]

\[
\text{ELTELE} := \text{SPACING IN IMAGE ELEMENTS BETWEEN CONSECUTIVE AREA ELEMENTS}
\]

THESE FOUR PARAMETERS MAY BE OBTAINED FROM THE AREA DIRECTORY ENTRY. WHEN \(\text{LINES} = \text{ELTELE} = 1\), THE AREA IS SAID TO BE AT "RESOLUTION 1", OR "FULL VISIBLE RESOLUTION" (SINCE THE GOES MODE AA VISIBLE CHANNEL IS CAPABLE OF A MAXIMUM RESOLUTION OF 1.). WHEN (AS ANOTHER EXAMPLE) \(\text{LINES} = \text{ELTELE} = 4\), ONLY EVERY FOURTH LINE AND ELEMENT ARE REPRESENTED IN THE AREA, AND THE AREA IS SAID TO BE AT "RESOLUTION 4", OR HAVE A "BLOW-DOWN FACTOR" OF 4 FROM RESOLUTION 1.

INTERNAL ORGANIZATION OF A DIGITAL AREA (COMMON FEATURES)

THE DATA RECORDED ON A TAPE FOR A SATELLITE IMAGE DISTRIBUTED BY SEEC IS AN EXACT COPY OF A DIGITAL AREA IN THE MCIDAS SYSTEM, SO IT IS NECESSARY TO KNOW THE INTERNAL STRUCTURE OF AREAS. EVERY AREA IS VIEWED AS A CONTINUOUS STREAM OF BYTES NUMBERED FROM 0. (EACH DATA BLOCK ON THE TAPE CONTAINS AN INDICATOR SHOWING WHERE THE BLOCK IS POSITIONED IN THIS CONTINUOUS STREAM.) WITHIN THE STREAM OF BYTES, THE AREA DATA IS PRESENTED LINE-BY-LINE, WITH THE FIRST LINE FIRST. EACH LINE IS FURTHER DIVIDED INTO
TWO PARTS. THE "LINE PREFIX" AND THE ACTUAL LINE DATA (IMAGE
ELEMENTS.) THE LINE PREFIX CONTAINS FURTHER DOCUMENTATION ABOUT
THE IMAGE AND THE GIVEN LINE, AND CAN BE ESSENTIALLY IGNORED
WHEN THAT INFORMATION IS NOT OF INTEREST.

THE SIZE OF THE PREFIX AND THE INFORMATION WITHIN IT
DEPEND HEAVILY ON THE AREA TYPE, WHICH IN TURN IS DETERMINED
BY THE DATA SOURCE. THE AREA TYPE IS GIVEN BY WORD 2 OF THE
AREA DIRECTORY ENTRY (Q.V.). REGARDLESS OF AREA TYPE,
EVERY LINE WITHIN THE AREA HAS THE SAME LENGTH PREFIX. THIS
LENGTH, IN BYTES, IS GIVEN BY WORD 15 OF THE AREA DIRECTORY
ENTRY. IT IS ALWAYS A MULTIPLE OF 4.

THE LINE PREFIX MAY OPTIONALLY BEGIN WITH A "VALIDITY CODE".
ITS PRESENCE IS SIGNIFIED BY A NON-ZERO VALUE IN WORD
36 OF THE AREA DIRECTORY ENTRY. WHEN PRESENT, THE VALIDITY
CODE OCCUPIES THE FIRST 4 BYTES OF THE LINE PREFIX. THESE
FIRST 4 BYTES MUST MATCH EXACTLY WORD 36 OF THE AREA DIRECTORY
ENTRY, OR ELSE THE LINE IS INVALID AS DATA AND MUST BE IGNORED.

THE LINE PREFIX IS FOLLOWED BY THE IMAGE ELEMENTS IN THAT
LINE. AN ELEMENT MAY BE AS SHORT AS 1 BYTE (SINGLE-BAND
IMAGES, WHICH HAVE 8-BIT PIXELS) OR AS LONG AS 24 BYTES (12
BANDS OF 16-BIT PIXELS.). THE NUMBER OF BYTES IN AN ELEMENT
IS GIVEN BY WORD 11 OF THE AREA DIRECTORY ENTRY.

EACH LINE IN AN AREA HAS THE SAME TOTAL LENGTH, AND THIS
LENGTH (IN BYTES) IS ALWAYS A MULTIPLE OF 4.

ADDITIONAL STRUCTURE OF GOES VISSR AREAS

THE GOES VISSR INSTRUMENT PRODUCES IMAGES IN TWO SPECTRAL
BANDS, DUBBED "VISIBLE" AND "INFRARED" (IR). A PARTICULAR AREA
CONTAINS ONLY ONE OF THESE TWO BANDS. WHICH BAND IT IS MAY BE
DETERMINED FROM THE SATELLITE CODE (SSS) IN WORD 3 OF THE
AREA DIRECTORY ENTRY. GENERALLY, EVEN-NUMBERED SSS REPRESENT VISIBLE
IMAGES, AND ODD-NUMBERED SSS REPRESENT IR IMAGES.

EVERY ELEMENT IN A VISSR AREA CONTAINS JUST ONE 8-BIT
PIXEL, REPRESENTING RAW DATA FROM THE INSTRUMENT. IF THE AREA
CONTAINS IR DATA, THE OBSERVED TEMPERATURE MAY BE DERIVED
FROM THE PIXEL VALUE, ACCORDING TO THE FOLLOWING FORMULAE:

\[ T = \frac{419 - B}{B/2} \] (B≤176 OR B=176)
\[ T = 330 - (B/2) \] (B>176 OR B=176)

WHERE:

- T IS BRIGHTNESS TEMPERATURE (DEGREES K)
- B IS PIXEL VALUE (0...255)

NOTE THAT FOR IR DATA, THE HIGHEST PIXEL VALUES CORRESPOND
TO THE COLDEST TEMPERATURES ("SPACE IS WHITE").

THE LINE PREFIX IN A VISSR AREA MAY BE ABSENT ENTIRELY.
OR MAY BE JUST 4 BYTES FOR A VALIDITY CODE, FOR IR AREAS ONLY.
THE LINE PREFIX MAY INCLUDE 126 BYTES FOR THE IR DOCUMENTATION
TRANSMITTED BY THE SATELLITE ALONG WITH THE DATA. BUT NOTE
THAT MOST OF THE USEFUL INFORMATION IN THE IR DOCUMENTATION
HAS BEEN PROCESSED BY SSEC INTO MORE USEFUL FORM IN NAVIGATION.
RECORDS, WHICH ARE INCLUDED ON TAPE ALONG WITH THE DATA.

THE HIGHEST RESOLUTION (I.E., LOWEST VALUES OF LINES)
AND ELAPSES IN THE AREA DIRECTORY ENTRY) POSSIBLE FOR A VISIBLE
AREA IS 1; FOR AN IR AREA IT IS ONLY 4 (LONGER WAVELENGTHS
HAVE INHERENTLY LESS RESOLUTION).

ADDITIONAL STRUCTURE OF VAS AREAS (SOUNDER AREAS)

THE VAS INSTRUMENT PRODUCES OBSERVATIONS IN UP TO
12 DIFFERENT IR SPECTRAL BANDS AT A GIVEN SPATIAL LOCATION.
PLUS A VISIBLE BAND. ALL OR SOME OF THE IR BANDS MAY BE
REPRESENTED TOGETHER IN A SINGLE AREA OF VAS TYPE; BUT
THE VISIBLE BAND IS REPRESENTED SEPARATELY IN AN AREA OF VISIR
TYPE. THUS, AS WITH GOES VISSR IMAGES, IT MAY REQUIRE A PAIR
OF AREAS TO OBTAIN THE TOTAL INFORMATION TRANSMITTED BY THE
SATELLITE DURING A GIVEN TIME PERIOD.

THE CHARACTERISTICS OF THE VAS BANDS ARE GIVEN FOLLOWS:

<table>
<thead>
<tr>
<th>BAND NUM.</th>
<th>CENTER WAVELENGTH (UM)</th>
<th>WIDTH (1/CM)</th>
<th>PURPOSE</th>
<th>MAIN ABSORBING GASES</th>
<th>OTHER EFFECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14.73</td>
<td>678.7</td>
<td>TEMP</td>
<td>CO₂</td>
<td>O₃</td>
</tr>
<tr>
<td>2</td>
<td>14.48</td>
<td>690.5</td>
<td>TEMP</td>
<td>CO₂</td>
<td>O₃</td>
</tr>
<tr>
<td>3</td>
<td>14.25</td>
<td>701.6</td>
<td>TEMP</td>
<td>CO₂</td>
<td>O₃</td>
</tr>
<tr>
<td>4</td>
<td>14.01</td>
<td>713.6</td>
<td>TEMP</td>
<td>CO₂</td>
<td>O₃</td>
</tr>
<tr>
<td>5</td>
<td>13.33</td>
<td>750.0</td>
<td>TEMP</td>
<td>CO₂, H₂O</td>
<td>O₃</td>
</tr>
<tr>
<td>6</td>
<td>4.525</td>
<td>2210.0</td>
<td>TEMP+ CLOUD</td>
<td>N₂O</td>
<td>SUN</td>
</tr>
<tr>
<td>7</td>
<td>12.66</td>
<td>790.0</td>
<td>MOISTURE</td>
<td>H₂O</td>
<td>CO₂+DUST</td>
</tr>
<tr>
<td>8</td>
<td>11.17</td>
<td>895.0</td>
<td>SURFACE</td>
<td>H₂O</td>
<td>CO₂+DUST</td>
</tr>
<tr>
<td>9</td>
<td>7.261</td>
<td>1377.2</td>
<td>MOISTURE</td>
<td>H₂O</td>
<td>CO₂</td>
</tr>
<tr>
<td>10</td>
<td>6.725</td>
<td>1487.0</td>
<td>MOISTURE</td>
<td>H₂O</td>
<td>---</td>
</tr>
<tr>
<td>11</td>
<td>4.444</td>
<td>2250.0</td>
<td>TEMP+ CLOUD</td>
<td>N₂O, CO₂</td>
<td>SUN</td>
</tr>
<tr>
<td>12</td>
<td>3.945</td>
<td>2535.0</td>
<td>SURFACE</td>
<td>H₂O</td>
<td>SUN+DUST</td>
</tr>
</tbody>
</table>
CONCEPTUALLY, MD FILES LOOK LIKE A LARGE 2-DIMENSIONAL MATRIX
OF DATA RECORDS. THESE RECORDS ARE ADDRESSED BY 2-D COORDINATES:
E.G. (3,4). (THE UPPER-LEFT-MOST RECORD IS (1,1).) ATTACHED
TO EACH ROW IS A LABEL-- AN ADDITIONAL RECORD OF SEVERAL WORDS--
KNOWN AS THE ROW HEADER. SIMILARLY, ATTACHED TO EACH COLUMN IS A
COLUMN HEADER. LOGICALLY SPEAKING, THE WHOLE FILE THUS LOOKS LIKE
A LARGE TABLE, WITH MULTIPLE ITEMS OF INFORMATION IN EACH BOX,
AND LABELS ATTACHED TO EACH ROW AND COLUMN. THE ACTUAL USEFUL
INFORMATION ASSOCIATED WITH EACH BOX CONSISTS OF THE INFORMATION
IN THE BOX, PLUS THE ASSOCIATED ROW AND COLUMN LABELS. USING THE
MD JARCON:

====> RECORD = ROW HEADER + COLUMN HEADER + DATA PORTION

IN MATRIX COORDINATES, THE ROW HEADERS ARE THOUGHT TO BE IN COLUMN
1, AND THE COLUMN HEADERS IN ROW 0. Thus:

====> RECORD = (X,Y) + (X,N) + (Y,N)

THE RECORD THUS DEFINED IS SIMPLY A GROUP OF WORDS (32-BIT
ENTITIES), IN THE ORDER JUST GIVEN. THE OTHER KEY CONCEPT OF MD
FILES IS THAT THE WORDS IN A RECORD ARE GIVEN NAMES, CALLED "KEYS"
THAT THESE NAMES ARE RECORDED IN ADVANCE, IN A LIST CALLED A FILE
"SCHEMA", AND THE ENTIRE SCHEMA IS GIVEN A NAME, WHICH IN A SENSE
DEFINES THE FILE TYPE. FOR EXAMPLE, THE SCHEMA DEFINING MD FILES
TO CONTAIN DATA FROM RDCONDE OBSERVATIONS IS CALLED "RA05", AND
CONSISTS OF KEYS SUCH AS "P" (PRESSURE), "T" (TEMPERATURE),
"TD" (DEWPOINT), AND SO FORTH. (ALL KEY NAMES ARE 1-4 LETTERS.)

ONCE DEFINED, THIS SCHEMA DESCRIBES THE DATA IN ALL MD FILES OF
THIS TYPE.

A USEFUL CONVENTION, WHICH SHOULD BE FOLLOWED WHENEVER THERE IS NO
COMPELLING REASON NOT TO, IS THAT SCHEMAS ARE DESIGNED SO THAT ALL
DATA FOR A PARTICULAR TIME IS GROUPED IN A SINGLE ROW, AND THE ROW
HEADER DESCRIBES THE TYPE (I.E., DATE AND TIME, PERHAPS SENSOR
TYPE, DATE, AND TIME); ALL DATA COLLECTED AT A PARTICULAR GEOGRAPHIC
LOCATION IS GROUPED IN A SINGLE COLUMN, AND THE COLUMN HEADER
DEScribes THE LOCATION (E.G., STATION #, ITS LATITUDE AND LONGI-
ITUDE, ELEVATION, ETC.). THEN THE TIME AND PLACE OF AN OBSER-
VATION CAN DETERMINE A UNIQUE ROW-COLUMN COORDINATE AT WHICH TO
PLUNK THE DATA COMPRISING THE OBSERVATION. THIS KIND OF
FILE ORGANIZATION KEEPS THE DATA FORMAT VERY CLOSE TO THE ACTUAL
STRUCTURE OF THE DATA, AND TO INTUITION.

SCHEMAS ARE PREPARED USING THE EDITOR; THEIR FORMAT WILL BE SPECIFIED
ELSEWHERE. ONCE PREPARED, THE SCHEMA MUST BE "REGISTERED" WITH THE
MD SYSTEM (USING A PROGRAM CALLED SCHEMA) BEFORE THEY CAN BE USED
TO CREATE NEW MD FILES.
THE MD FILES ARE IMPLEMENTED AS LW ("LARGE WORD ARRAY") LEVEL FILES USING THE LW-PREFIX SUBROUTINES. IN OTHER WORDS, THE MATRIX-OF-RECORDS STRUCTURE THAT DISTINGUISHES THE MD LEVEL IS SUPERIMPOSED ON A STRUCTURE THAT PRESENTS A FILE AS A SIMPLE CONTIGUOUS ARRAY OF WORDS. THE LOCATION OF THE VARIOUS MD FILE COMPONENTS IN THE CONTIGUOUS-WORD SPACE ARE GIVEN BELOW (ALL WORD #'S ARE 0-BASED).

FIRST 4095 WORDS:

FIRST 64 WORDS ARE THE FILE HEADER (LABEL).

NEXT 400 WORDS ARE THE USER RECORD (MD COORDINATES 0,0).

THIS BLOCK OF WORDS IS NOT DESCRIBED BY ANY PART OF THE SCHEMA AND IS AVAILABLE FOR STORAGE OF Arbitrary INFORMATION.

NEXT 400 WORDS CONTAIN THE NAMES OF THE FILE KEYS.

NEXT 400 WORDS CONTAIN THE SCALE FACTORS OF THE KEYS.

NEXT 400 WORDS CONTAIN THE UNITS OF THE KEYS.

ALL WORDS BETWEEN HERE AND WORD # 4095 ARE RESERVED FOR LATER EXPANSION OF THE SCHEMA. DO NOT USE THIS AREA FOR ANYTHING.

BEGINNING AT WORD # 4096:

FIRST 300 WORDS CONTAIN ALL THE ROW HEADERS.

NEXT 100 WORDS CONTAIN ALL THE COLUMN HEADERS.

NEXT 100 WORDS CONTAIN THE DATA PORTION: THE MAIN MATRIX.

THE DATA RECORDS ARE STORED ROW-BY-ROW, SO ROW-BY-ROW ACCESS WILL PROVE MOST EFFICIENT.

THE NEXT WORD AFTER HERE IS MARKED BY FENDF IN MD.COM. ALL STORAGE BEYOND THIS POINT IS AVAILABLE FOR USER PURPOSES.
THIS DOCUMENT DESCRIBES THE STRUCTURE OF SOUNCING FILES ON IBM
MC-IDAS SYSTEMS.

SOUNCING FILES ON THE IBM ARE ONE OF THE SUBTYPES OF THE AREA
SYSTEM. THAT IS THEY HAVE A LINE ORIENTED STRUCTURE AND ARE
ACCESSED WITH SAME ROUTINES (E.G. OPNA, READA) AS AREAS. THE
DIFFERENCES ARE:
1) A POINT IN IMAGE SPACE HAS MORE THAN ONE ASSOCIATED VALUE
2) VALUES ARE SIGNED 2 BYTE INTEGERS RATHER THAN UNSIGNED BYTES
3) THERE IS A LINE HEADER FOR EACH LINE

THE ORGANIZATION OF THE FILES IS SIMILAR TO THAT OF SOUNCING
FILES ON THE HARRIS WITH THE EXCEPTION THAT THE LINE HEADER IS
ORGANIZED DIFFERENTLY. THE DIFFERENCE IS THAT ALL THE CHANNEL
SPECIFIC INFORMATION FOR A CHANNEL IS CONTIGUOUS. THERE ARE FOUR
DATA ITEMS FOR EACH CHANNEL WHICH ARE, IN ORDER, CHANNEL NUMBER,
NUMBER OF SPINS, DELTA F, AND Y SUE Z. EACH OF THESE CAN BE
CONSIDERED A SIGNED 2 BYTE QUANTITY.
USER COMMON IS ACCESSED BY SUBROUTINES *LUC* (LOOK AT) AND
*PUC* (PUT INTO).

UC VARIABLES RESIDE IN THE SYSTEM COMMUNICATIONS REGION (SYSCOM):
they are not lost when your initiator aborts, but only when MCIOS
ABORTS. UC VARIABLES WITH NEGATIVE SUBSCRIPTS Pertain to the INITI-
ATOR AND RESIDE IN THE INITIATOR REGION OF SYSCOM (THIS INCLUDES
UC(0)); UC VARIABLES WITH POSITIVE SUBSCRIPTS Pertain to YOUR TERMINAL
AND RESIDE IN THE (SEPARATE) TERMINAL REGION OF SYSCOM. (ALL THIS IS
TRANSPARENT TO ANY PROGRAM WHICH CALLS *LUC* OR *PUC*.)

PRECEDING THE UC VARIABLES IN SYSCOM IS THE GLOBAL SYSCOM AREA,
WHICH CONTAINS VARIABLES ACCESSIBLE TO ALL INITIATORS AND AT ALL
TERMINALS. THESE VARIABLES ARE INSPECTED BY FUNCTION WD(IPOS)
AND WRITTEN BY SUBROUTINE WD(IPOS,IVAL). WD(0) RETURNS THE FIRST
WORD OF THIS AREA.

LAYOUT OF THE GLOBAL SYSCOM REGION

(INDICES ARE THOSE USED TO ACCESS SYSCOM VIA SUBPROGRAMS ND/WD)

1-2) SYSTEM VERSION.
3) YYDDD WHEN *WAKEUP* WAS EXECUTED
4) PHMSS WHEN *WAKEUP* WAS EXECUTED
5) LARGEST FOREGROUND TERMINAL NUMBER DEFINED IN SYSTEM
   (184: CURRENT NUMBER IS 71)
6) MAXIMUM PERMITTED AREA NUMBER (BETWEEN 1 AND 9999)
7-299) RESERVED

LAYOUT OF UC

THE FOLLOWING WORDS GIVE THE NAMES OF NAVIGATION CORELOADS WHICH
HAVE BEEN DYNAMICALLY LOADED INTO CORE. THESE ARE REFERENCED
BY SUBROUTINE VLOPE, WHICH RELEASES THE CORE USED BY THESE
CORELOADS AT THE CONCLUSION OF EACH SQX*ED MODULE.

-94, -95) RESERVED FOR FUTURE NAVIGATION CORELOADS
-93) SUFFIX OF NAVIGATION-3 CORELOAD (NAME=*NV3* + SUFFIX)
   (E.G. *GOFS*)
-92) SUFFIX OF NAVIGATION-2 CORELOAD (NAME=*NV2* + SUFFIX)
-91) SUFFIX OF NAVIGATION-1 CORELOAD (NAME=*NV1* + SUFFIX)
DATA TABLET GROUP. THESE WORDS CONTAIN THE INFORMATION
RETURNED FROM THE TERMINAL ON THE TABLET STATE, AS WELL AS
THE TABLET BOX NUMBER, NAME AND POSITION AS DETERMINED BY
THE SCANNER.

-79) HIGHEST Y COORDINATE OF BOX (TOP)
-78) LOWEST Y COORDINATE OF BOX (BOTTOM)
-77) HIGHEST X COORDINATE OF BOX (RIGHT)
-76) LOWEST X COORDINATE OF BOX (LEFT)
-75) BOX STRING NAME (A4 EBCDIC)
-74) BOX NUMBER (BOX 1 IS AT UPPER LEFT)
-73) PEN MODE (2=CR 3=pen down, 1=approximate, 0=pen up)
-72) PEN POSITION-- Y COORDINATE: (0,0) IS BOTTOM-LEFT
-71) PEN POSITION-- X COORDINATE

ACTION VECTORE (WINDCO) UUCWORDS USED (SEE ALSO UC 99,100):

-82) JOYSTICK CONTROL VALUE SAVE
-81) MODE CONTROL FOR JOYSTICK 2
-80) MODE CONTROL FOR JOYSTICK 1
-79) ENDING ROW NUMBER IN CURRENT MD 'WIND' FILE
-78) ENDING COLUMN NUMBER IN CURRENT MD 'WIND' FILE
-77) STARTING ROW NUMBER IN CURRENT MD 'WIND' FILE
-76) CURRENT MD 'WIND' FILE USED FOR CORE OUTPUT
-75) CURRENT MD 'WIND' FILE USED INV. SELECTOR AND
-74) CORE INPUT
-73) ELEMENT SIZE OF TARGET CURSOR (TV COORD)
-72) LINE SIZE OF TARGET CURSOR (TV COORD)
-71) SPACE BAR TOGGLE
-70) ERROR STATUS WORD

-64) 2ND CURSOR STATE CONTROL WORD

FORMAT:
BIT 0-2 NOT USED
BIT 3-5 ARE CURSOR COLOR (DT COMMAND)
BIT 6-7 ARE CURSOR TYPE (DT COMMAND)

-63) 2ND CURSOR ELEMENT POSITION
-62) 2ND CURSOR LINE POSITION
-61) 2ND CURSOR ELEMENT SIZE
-60) 2ND CURSOR LINE SIZE

-54) CURRENT GRAPHICS VIRTUAL FRAME
-53) CURRENT EMS DEVICE: 1(CRT), 2(LOCAL PRTR), 3(SYS PRTR), 0(BLACK HOLE)
-52) CURRENT EMS DEVICE: 1(CRT), 2(LOCAL PRTR), 3(SYS PRTR), 0(BLACK HOLE)
-51) CURRENT LTO DEVICE: 1(CRT), 2(LOCAL PRTR), 3(SYS PRTR), 0(BLACK HOLE)

-30) AUTO-CONTEXT-TABLE-SEARCH: WHEN 1, PARAMETER FETCHING SUBRS. (E.G.
-31) CKWP) REFERS TO SYSTEM STRING TABLE FOR MISSING KEYWORD PARAMS

-29) (RESERVED FOR FUTURE USE BY SYSTEM SUBROUTINE SQY)
-28) NAME (CHARS 5-8) OF PROGRAM CURRENTLY RUNNING
-27) NAME (FIRST 4 CHAR.) OF PROGRAM CURRENTLY RUNNING
-26) PROGRAM CURRENTLY RUNNING IS MACRO (=1), NON-MACRO (=0)
-25) PROGRAM CURRENTLY RUNNING IS FOREGROUND (=1), BACKGROUND (=0)
-24) BACKGROUND W/ FORTRAN MAIN: CAN DO FORTRAN I/O (=1)
-23) CONSOLE STARTED TASK I.E. TAPE JOB (=2)
-22) CURRENT SIX LEVEL'S (SCANNER IS 0, NEXT PGM IS 1, ETC.)
-21) NUMBER OF THIS INITIATOR
-20) KEYIN WAS STARTED BY SCHEDULER (=1); NOT BY SCHEDULER (=0)
-19) RESERVED FOR LATER USE
-18) USER INITIALS UNDER WHICH THIS KEYIN RUNS (MAY BE DIFFERENT FROM LOGGED-ON INITIALS IN UC(2) IF KEYIN IS SCHEDULED)
-17) PROJECT # UNDER WHICH THIS KEYIN RUNS (MAY BE DIFFERENT FROM LOGGED-ON PROJECT #: IN UC(1))
-16) CONTAIN A SNAPSHOT OF THE TERMINAL STATE TAKEN JUST BEFORE THE KEYIN BEGINS. THIS IS IN A FORMAT CLOSE TO THE TERMINAL PROTOCOL.

-13) CURSOR STATE CONTROL WORD

124) FORMAT: BIT 0-2=0 MEANS CURSOR FROZEN (SET BY PC COMMAND)
125) BIT 0-2=1 MEANS CURSOR POS IS JOYSTICK-CONTROLLED (P KEY)
126) BIT 0-2=2 MEANS CURSOR SIZE & POS ARE JOYSTICK-CNTRLD (Z KEY)
127) BIT 3-5 ARE CURSOR COLOR (D T COMMAND)
128) BIT 6-8 ARE CURSOR TYPE (D T COMMAND)
129) FORMAT 7/83: BIT 0-2 CONTROL JOYSTICK ASSIGNMENTS:
130) 0--CURSOR IS HOST-CONTROLLED
131) 1--JOYSTICK 1 IS POS VERNIER & J2 IS COARSE POS
132) 2--J1 IS CURSOR SIZE & J2 IS DISCONNECTED
133) 3--J1 IS CURSOR SIZE & J2 IS COARSE POSITION
134) 4--J1 IS VELOCITY & J2 IS DISCONNECTED
135) 5--J1 IS VELOCITY & J2 IS COARSE POSITION
136) 6--DUAL CURSOR MODE: HOST CONTROLLED
137) 7--DUAL CURSOR MODE: J1 IS VERNIER POSITION OF 2ND CURSOR
138) WHILE J2 IS COARSE POSITIONS OF BOTH CURSORS

139) CURSOR ELEMENT POSITION
140) CURSOR LINES POSITION
141) CURSOR LINES SIZE
142) GRAPHICS STATE CONTROL WORD

143) FORMAT: BIT 0=1 MEANS GRAPHICS CONNECTED TO LOOP CONTROL (J KEY)
144) BIT 1=1 MEANS GRAPHICS FRAME IS LOOPING (L KEY)
145) BIT 2=1 MEANS GRAPHICS FRAME IS BLANKED (W KEY)

146) GRAPHICS LOWER ROUND
147) GRAPHICS UPPER ROUND
148) CURRENT GRAPHICS FRAME
149) FRAME STATE CONTROL WORD

150) FORMAT: BIT 0=1 MEANS FRAME CONNECTED TO LOOP CONTROL (Y KEY)
151) BIT 1=1 MEANS FRAME IS LOOPING (L KEY)
152) BIT 2=1 MEANS FRAME IS BLANKED (W KEY)
153) IMAGE FRAMES LOWER ROUND
154) IMAGE FRAMES UPPER ROUND
155) CURRENT IMAGE FRAME
156)
17. User's terminal number

18. Project < Memo which current user is logged on (may be different from UC(-15))

19. User's initials

20. Current nav file

21. Current no file

22. Current grid file

23. Second printer number for this user (SYS PRTR=0)

24. First printer number (local printer for this user; SYS PRTR=0)

25. # of lines on TV screen

26. # of elements on TV screen

27. # of image frames

28. # of graphics frames

29. Terminal is remote (=1); local (=0)

30. Terminal is video (=1); non-video (=0)

31. Terminal has received mods of 7/13 (=0); has not (=1)

32. (This affects the operation of the key program)

33. SVCT flag on the terminal wanting 604 over comm

34. MCIAS Disk File I/O Trace: 1=DO Trace, anything else=DO NOT

35. Set to 1 by "G" key

36. Set to 1 by "9" key

37. Import pointer for others...JMB

38. Terminal identifier (name or location)--8 characters

39. Name of CRT component of terminal--4 characters

40. (*TLVC* TELEVIDEO *3278*=IBM 3278 *CRT=OTHERS)

41. Data tablet--horizontal extent (max x coordinate + 1)

42. Data tablet--vertical extent (max y coordinate + 1)

43. Non-zero => terminal down

44. Non-zero => may quit areas even if not the owning project

45. (Operator's terminal only)

46. Draw graphics on WRRM* 1=DO NOT draw graphics on WRRM

47. 0=DO NOT WRITE VIRTUAL GRAPHIC; NON-ZERO=VIRTUAL FRAME TO WRITE

48. 3-D GRAPHICS MENU POINTER

49. Planetary Programs Navigation Pointer

50. Flipak Right/Left Frame Inversion flag for two screen stereo

51. Reserved for 3-D Graphics and Planetary Programs Status File ID

52. Reserved for 3-D Graphics and Planetary Programs

53. 3-D Graphics Message Waiting Flag

54. Default Graphics Line Width

55. Graphics Dash Pattern (1)

56. Graphics Dash Pattern (2)

57. Graphics Dash Pattern (3)

58. Words 60...74 are reserved for keeping the host's copy of the terminal state

59. This is in a format easy to manipulate and related to the functions of the terminal as an abstract device.

60. Subroutine STATUS COMPOSES THIS STATE INTO A VALID TERMINAL ORDER (PROTOCOL RESEMBLES SNAPSHOT IN UC -1...-13)

61. Loop control: 1 (system is looping) 3 (not looping)

62. Loop control system consists of L=command & A, R, J, Y, O & L keys
IMAGES & GRAPhICS ARE INDEPENDENTLY CONNECTED TO & DISCONNECTED FROM THE LOOP SYSTEM VIA UC 54 AND 59.

1) CURRENT IMAGE FRAME

2) IMAGE FRAME LOOP-- UPPER BOUND

3) IMAGE FRAME LOOP-- LOWER BOUND

4) 1 (IMAGE FRAMES CONNECTED TO LOOP CONTROL) 0 (NOT CONNECTED)

5) 1 (IMAGE FRAMES VISIBLE) 0 (IMAGE FRAMES BLANKED)

6) CURRENT GRAPhICS FRAME

7) GRAPhICS FRAME LOOP-- UPPER BOUND

8) GRAPhICS FRAME LOOP-- LOWER BOUND

9) 1 (GRAPhICS FRAMES CONNECTED TO LOOP CONTROL) 0 (NOT CONNECTED)

10) 1 (GRAPhICS FRAMES VISIBLE) 0 (BLANKED)

11) CURSOR SIZE (VERTICAL)

12) CURSOR SIZE (HORIZONTAL)

13) CURSOR POSITION (LINE #)

14) CURSOR POSITION (ELEMENT #)

15) CURSOR TYPE (BOX) 2 (CROSSHAIR) 3 (BOX&CROSSHAIR) 4 (SOLID BOX)

16) CURSOR TYPE 5 (STARS WARS)

17) 1 (DISCONNECTED) 0 (CONNECTED)

18) 1 (CONTROLS CURSOR POSITION)

19) 3 (CONTROLS CURSOR SIZE)

20) 4 (VELOCITY CURSOR)

21) 1 (SAME VALUES AS JOYSTICK 1)

22) CURRENTLY THE ONLY LEGAL STATES FOR JOYSTICK1/JOYSTICK2 ARE:

0/0 2/1 3/1 3/0 4/1 4/0 IN SINGLE CURSOR MODE

0/0 2/1 IN DUAL CURSOR MODE

23) 1 (SATURDAY) 1 (SUNDAY)

24) 1 (MONDAY) 1 (TUESDAY)

25) 1 (WEDNESDAY) 1 (THURSDAY)

26) 1 (FRIDAY) 1 (SATURDAY)

27) 1 (SUNDAY) 1 (MONDAY)

28) 1 (TUESDAY) 1 (WEDNESDAY)

29) 1 (THURSDAY) 1 (FRIDAY)

30) 1 (SATURDAY) 1 (SUNDAY)

31) 1 (MONDAY) 1 (TUESDAY)

32) 1 (WEDNESDAY) 1 (THURSDAY)

33) 1 (FRIDAY) 1 (SATURDAY)

34) 1 (SUNDAY) 1 (MONDAY)

35) 1 (TUESDAY) 1 (WEDNESDAY)

36) 1 (THURSDAY) 1 (FRIDAY)

37) 1 (SATURDAY) 1 (SUNDAY)

38) 1 (MONDAY) 1 (TUESDAY)

39) 1 (WEDNESDAY) 1 (THURSDAY)

40) 1 (FRIDAY) 1 (SATURDAY)

41) 1 (SUNDAY) 1 (MONDAY)

42) 1 (TUESDAY) 1 (WEDNESDAY)

43) 1 (THURSDAY) 1 (FRIDAY)

44) 1 (SATURDAY) 1 (SUNDAY)

45) 1 (MONDAY) 1 (TUESDAY)

46) 1 (WEDNESDAY) 1 (THURSDAY)

47) 1 (FRIDAY) 1 (SATURDAY)

48) 1 (SUNDAY) 1 (MONDAY)

49) 1 (TUESDAY) 1 (WEDNESDAY)

50) 1 (THURSDAY) 1 (FRIDAY)

51) 1 (SATURDAY) 1 (SUNDAY)

52) 1 (MONDAY) 1 (TUESDAY)

53) 1 (WEDNESDAY) 1 (THURSDAY)

54) 1 (FRIDAY) 1 (SATURDAY)

55) 1 (SUNDAY) 1 (MONDAY)

56) 1 (TUESDAY) 1 (WEDNESDAY)

57) 1 (THURSDAY) 1 (FRIDAY)

58) 1 (SATURDAY) 1 (SUNDAY)

59) 1 (MONDAY) 1 (TUESDAY)

60) 1 (WEDNESDAY) 1 (THURSDAY)

61) 1 (FRIDAY) 1 (SATURDAY)

62) 1 (SUNDAY) 1 (MONDAY)

63) 1 (TUESDAY) 1 (WEDNESDAY)

64) 1 (THURSDAY) 1 (FRIDAY)

65) 1 (SATURDAY) 1 (SUNDAY)

66) 1 (MONDAY) 1 (TUESDAY)

67) 1 (WEDNESDAY) 1 (THURSDAY)

68) 1 (FRIDAY) 1 (SATURDAY)

69) 1 (SUNDAY) 1 (MONDAY)

70) 1 (TUESDAY) 1 (WEDNESDAY)

71) 1 (THURSDAY) 1 (FRIDAY)

72) 1 (SATURDAY) 1 (SUNDAY)

73) 1 (MONDAY) 1 (TUESDAY)

74) 1 (WEDNESDAY) 1 (THURSDAY)

75) 1 (FRIDAY) 1 (SATURDAY)

76) 1 (SUNDAY) 1 (MONDAY)

77) 1 (TUESDAY) 1 (WEDNESDAY)

78) 1 (THURSDAY) 1 (FRIDAY)

79) 1 (SATURDAY) 1 (SUNDAY)

80) 1 (MONDAY) 1 (TUESDAY)

81) 1 (WEDNESDAY) 1 (THURSDAY)

82) 1 (FRIDAY) 1 (SATURDAY)

83) 1 (SUNDAY) 1 (MONDAY)

84) 1 (TUESDAY) 1 (WEDNESDAY)

85) 1 (THURSDAY) 1 (FRIDAY)

86) 1 (SATURDAY) 1 (SUNDAY)

87) 1 (MONDAY) 1 (TUESDAY)

88) 1 (WEDNESDAY) 1 (THURSDAY)

89) 1 (FRIDAY) 1 (SATURDAY)

90) 1 (SUNDAY) 1 (MONDAY)

91) 1 (TUESDAY) 1 (WEDNESDAY)

92) 1 (THURSDAY) 1 (FRIDAY)

93) 1 (SATURDAY) 1 (SUNDAY)

94) 1 (MONDAY) 1 (TUESDAY)

95) 1 (WEDNESDAY) 1 (THURSDAY)

96) 1 (FRIDAY) 1 (SATURDAY)

97) 1 (SUNDAY) 1 (MONDAY)

98) 1 (TUESDAY) 1 (WEDNESDAY)

99) 1 (THURSDAY) 1 (FRIDAY)

100) CURRENT TVS AREA/SCANNER AREA/RTVL FILE #

101) CURRENT TVS AREA/SCANNER AREA/RTVL FILE #

DATA TABLET RELATED SETUP OPTIONS

102) TABLET- TV SPACE MODE

103) TABLET- INACTIVE AREA (BORDER) AROUND OUTSIDE

104) TABLET- CURSOR FOLLOWING STATE

105) TABLET- LOWER RIGHT CORNER OF TV SPACE (ELEMENT POSITION)

106) TABLET- LOWER RIGHT CORNER OF TV SPACE (ELEMENT POSITION)

107) TABLET- WHEN TO START SIGNIFICANT EVENT

108) TABLET- WHAT TYPE OF EVENT TO START

109) CURRENT MOTION VECTOR OUTPUT FILE FOR MOTION VECTORS (WINOCO)

110) CURRENT MOTION VECTOR OUTPUT FILE FOR MOTION VECTORS (WINOCO)
CALLING SEQUENCES

I=LUC(INDEX)

C

C------PEEK/POKE USER COMMON

C

C

C------LUC "LOOKS AT" USER COMMON (PEEK)

C------INDEX IS WHERE (IN USER COM.) TO DO TRANSFER

C------FN VAL IS WORD EXTRACTED FROM USER COMMON

C

C

C

I=PUC(FROM,INDEX)

OR: CALL PUC(FROM,INDEX) (NO FUNCTION VALUE)

C

C------PUC POKES WORD INTO USER COMMON.

C------INDEX AS IN LUC

C------FROM IS FULLWORD DATA VALUE TO TRANSFER TO USER COM.

C------FN VAL IS (ALSO) WORD TRANSFERRED

C

C
CONTEXT

SSYMD (SAT-YR-DAY OF YEAR)
HHMMSS (TIME AT BEGINNING OF SOUNDER FILE)
UPPER LEFT LINE NUMBER IN SATELLITE COORDINATES
UPPER LEFT ELEMENT NUMBER IN SATELLITE COORDINATES
UPPER LEFT HAND Z COORDINATE
NUMBER OF LINES IN IMAGE
NUMBER OF ELEMENTS IN IMAGE
NUMBER OF BYTES IN IMAGE
LINE RESOLUTION
ELEMENT RESOLUTION
Z RESOLUTION
CHANNELS (BANDS) AVAILABLE, 1 = YES
LLNW (LAT/LON AT NORTHWEST CORNER OF IMAGE)
LLSE (LAT/LON AT SOUTHEAST CORNER OF IMAGE)
DDHHMM OF SURFACE DATA USED
STATUS WORD
SURFACE GRID FILE NUMBER
CRIC NUMBERS FOR SURFACE ANALYSES (Z,T,TD,P)
GUESS PROFILE IDENTIFICATION
SOUNDER FILE NUMBER
SURFACE EDIT FILE MD NUMBER
SURFACE EDIT FILE ROW NUMBER
GUESS FILE MD NUMBER
GUESS FILE MD ROW NUMBER
RETRIEVAL FILE MD NUMBER
RETRIEVAL FILE MD ROW NUMBER
GUESS GRID FILE NUMBER
VISIBLE AREA NO. (CORRESPONDS TO SOUNDER FILE)
RETRIEVAL TYPE
FIRST GUESS
RETRIEVAL SPACING
FOV BOX SIZE
MOFF OPTION
LAST LINE
LAST ELEMENT
BEGIN LINE
BEGIN ELEMENT
TERMINAL NUMBER
PLOT OPTION
BIAS VALUES FOR CHAN RADIANCES
NUMBER OF RETRIEVALS
APPENDIX II

Level I Flowchart Constructs

The following brief appendix contains examples and (or) descriptions of all the major FORTRAN constructs used in the VAS program Level I Flowcharts discussed in this manual. I think most people with FORTRAN experience will find this appendix self-explanatory.
1. Offpage connector or beginning and end of DO-loops.

2. O-page connector or beginning and end of DO-loops.

3. Comment statement.

4. IF conditional.

5. CALL statement or arithmetic statement.

6. END statement.

7. DO-loop construct; loops back to beginning when \( A \) is reached in flow. If \( A \) is used, it means the loop extends past the end of the page. Note that loops do not always extend vertically; they can extend horizontally as well.

   Note: If a DO-loop extends beyond the end of the page, and there are jumps to the end of the DO-loop on the subsequent page where the loop terminates, then these jumps will be denoted by circles rather than the semi-box notation.

   When leaving DO-loop \( A \), control passes beyond last \( A \). This also holds for DO-loops with off-page connectors, in which case control would pass beyond the last \( A \).

8. Note that F (false) paths are NOT ALWAYS at the bottom of IF conditionals. Occasionally, the false path is to either the right or left of the conditional.

9. Numbers outside IF constructs, DO-loops, etc., indicate transfer of control to that particular statement of the program.
APPENDIX III

Subroutine List and Descriptions

The following appendix contains, in alphabetical order, information concerning the vast majority of the subroutines called by VAS retrieval programs. Included are all the Level I Subroutines, as well as further subservient subroutines. In particular, each subroutine included in the Chapter 2 Modular Flowcharts is listed and described here.

As part of each subroutine's description, the prefix or program/subroutine location within the McIDAS source library of each routine is listed. If the reader is referred to a subroutine/program location, it means the subroutine in question is located in its entirety somewhere within the designated subroutine/program library member. For example, the code for subroutine VTRET exists after the code for program SRET in library member VLSRET.

In addition, a brief discussion of what each subroutine does is also presented, along with descriptions of input arguments and/or output parameters and lists thereof whenever possible. Finally, the reader may occasionally be referred to the source code itself for further information.
<table>
<thead>
<tr>
<th>II</th>
<th>(MR)</th>
<th>GETFRM</th>
<th>(MR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>(SR)</td>
<td>GETGAM</td>
<td>(MR)</td>
</tr>
<tr>
<td>ANGGET</td>
<td>(VR)</td>
<td>GETNAV</td>
<td>(MR)</td>
</tr>
<tr>
<td>ANGLES</td>
<td>(MR)</td>
<td>GETSFV</td>
<td>(VR)</td>
</tr>
<tr>
<td>ANGSS</td>
<td>(in VLBNA)</td>
<td>GETTIM</td>
<td>(in MRGBETDAY)</td>
</tr>
<tr>
<td>ARASIZ</td>
<td>(MR)</td>
<td>GRADWI</td>
<td>(in VLGWVA)</td>
</tr>
<tr>
<td>ATOE</td>
<td>(MR)</td>
<td>H2OTAV</td>
<td>(VR)</td>
</tr>
<tr>
<td>BARB</td>
<td>(VR)</td>
<td>HEAPPY</td>
<td>(in MRFBARN)</td>
</tr>
<tr>
<td>BLKA</td>
<td>(SR)</td>
<td>HRTOP0</td>
<td>(TR)</td>
</tr>
<tr>
<td>BOX</td>
<td>(in MRPLTPBK)</td>
<td>HTV</td>
<td>(VR)</td>
</tr>
<tr>
<td>CALDAY</td>
<td>(VR)</td>
<td>HTX</td>
<td>(in VLSRET)</td>
</tr>
<tr>
<td>CLEANA</td>
<td>(SR)</td>
<td>IGNAME</td>
<td>(in MRGBIAKE)</td>
</tr>
<tr>
<td>CLEANW</td>
<td>(SR)</td>
<td>INITPL</td>
<td>(in MRPLTPBK)</td>
</tr>
<tr>
<td>CLMGES</td>
<td>(TR)</td>
<td>INTER</td>
<td>(in MRFBARN)</td>
</tr>
<tr>
<td>CLOSA</td>
<td>(in MRPOPNA)</td>
<td>INTPTW</td>
<td>(TR)</td>
</tr>
<tr>
<td>CLOSAO</td>
<td>(in MRRWITA)</td>
<td>IQ</td>
<td>(VR)</td>
</tr>
<tr>
<td>CO2TAV</td>
<td>(VR)</td>
<td>ISQX</td>
<td>(MR)</td>
</tr>
<tr>
<td>CONTAV</td>
<td>(VR)</td>
<td>ITOC</td>
<td>(SR)</td>
</tr>
<tr>
<td>CONTNT</td>
<td>(VR)</td>
<td>JMBWTF</td>
<td>(foreground version in MRGBWTF)</td>
</tr>
<tr>
<td>CSRAOB</td>
<td>(MR)</td>
<td>JSQX</td>
<td>(VR)</td>
</tr>
<tr>
<td>CSRIO</td>
<td>(in MRCRIO)</td>
<td>LINFIL</td>
<td>(in TRFILL)</td>
</tr>
<tr>
<td>CSRIO</td>
<td>(in MRCRIO)</td>
<td>LOCK</td>
<td>(SR)</td>
</tr>
<tr>
<td>CSRIO</td>
<td>(in MRCRIO)</td>
<td>LTQ</td>
<td>(MR)</td>
</tr>
<tr>
<td>CSRIO</td>
<td>(in MRCRIO)</td>
<td>LWCLS</td>
<td>(in MRLWGETX)</td>
</tr>
<tr>
<td>CSRIO</td>
<td>(in MRCRIO)</td>
<td>LWGET</td>
<td>(in MRLWGETX)</td>
</tr>
<tr>
<td>CSRIO</td>
<td>(in MRCRIO)</td>
<td>LWMO</td>
<td>(in MRLWSUBS)</td>
</tr>
<tr>
<td>CSRIO</td>
<td>(in MRCRIO)</td>
<td>LWNWF</td>
<td>(in MRLWSUBS)</td>
</tr>
<tr>
<td>CSRIO</td>
<td>(in MRCRIO)</td>
<td>LWOPEN</td>
<td>(in MRLWGETX)</td>
</tr>
<tr>
<td>CSRIO</td>
<td>(in MRCRIO)</td>
<td>LWPO</td>
<td>(in MRLWSUBS)</td>
</tr>
<tr>
<td>CSRIO</td>
<td>(in MRCRIO)</td>
<td>LWSO</td>
<td>(one version in MRS2LWS1)</td>
</tr>
<tr>
<td>DCLOSE</td>
<td>(in VRDAIO)</td>
<td>MDCLS</td>
<td>(in MRMDBMAKE)</td>
</tr>
<tr>
<td>DDEST</td>
<td>(MR)</td>
<td>MDNAME</td>
<td>(in MRMDBMAKE)</td>
</tr>
<tr>
<td>DIRADJ</td>
<td>(VR)</td>
<td>MNRAOB</td>
<td>(MR)</td>
</tr>
<tr>
<td>DOPEN</td>
<td>(in VRDAIO)</td>
<td>MOVB</td>
<td>(SR)</td>
</tr>
<tr>
<td>DREAD</td>
<td>(in VRDAIO)</td>
<td>MOVC</td>
<td>(SR)</td>
</tr>
<tr>
<td>DSHOFF</td>
<td>(in MRPLTPBK)</td>
<td>MOVW</td>
<td>(SR)</td>
</tr>
<tr>
<td>DSHON</td>
<td>(in MRPLTPBK)</td>
<td>MOVWC</td>
<td>(in SROVCW)</td>
</tr>
<tr>
<td>DWRITE</td>
<td>(in VRDAIO)</td>
<td>NSTAR</td>
<td>(in VLSRAD)</td>
</tr>
<tr>
<td>EDEST</td>
<td>(MR)</td>
<td>NVINIT</td>
<td>(VR)</td>
</tr>
<tr>
<td>EMES</td>
<td>(MR)</td>
<td>O3TAV</td>
<td>(VR)</td>
</tr>
<tr>
<td>ENAREA</td>
<td>(MR)</td>
<td>OPNA</td>
<td>(MR)</td>
</tr>
<tr>
<td>ENCODE</td>
<td>(VR)</td>
<td>OUTINT</td>
<td>(MR)</td>
</tr>
<tr>
<td>ENCODX</td>
<td>(TR)</td>
<td>PACK</td>
<td>(in SRCRACK)</td>
</tr>
<tr>
<td>ENPT</td>
<td>(in MRPLTPBK)</td>
<td>PAGE</td>
<td>(in MRPLTPBK)</td>
</tr>
<tr>
<td>EPOCH</td>
<td>(in MRGETNAV)</td>
<td>PENADD</td>
<td>(in MRPLTPBK)</td>
</tr>
<tr>
<td>EXTEMP</td>
<td>(TR)</td>
<td>PENBEG</td>
<td>(in MRPLTPBK)</td>
</tr>
<tr>
<td>FBARN</td>
<td>(MR)</td>
<td>PENMOV</td>
<td>(in MRPLTPBK)</td>
</tr>
<tr>
<td>FILC</td>
<td>(in VLSRAD)</td>
<td>PLNIV</td>
<td>(VR)</td>
</tr>
<tr>
<td>FILCLR</td>
<td>(in VLSRAD)</td>
<td>PLOT</td>
<td>(in MRPLTPBK)</td>
</tr>
<tr>
<td>FILL</td>
<td>(TR)</td>
<td>PLTDIG</td>
<td>(MR)</td>
</tr>
<tr>
<td>FILTER</td>
<td>(in VLVTZ)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLOC</td>
<td>(in VLXRA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GANTAV</td>
<td>(VR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GESPRO</td>
<td>(TR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETDAY</td>
<td>(MR)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
POST (SR) UNLOCK (in SRLOCK)
PRCL (in SRPRRD) VALUE (in VLSRVA)
PRCLOS (in SRPROPEN) VASDAT (VR)
PREATV (VR) VASDIG (VR)
PRECW (TR) VASGES (VR)
PRETAV (VR) VASNAV (in MRGETGAM)
PRLINX (SR) VASRTE (VR)
PROFIX (TR) VASTAU (VR)
PROPEN (SR) VASTPW (VR)
PRPRPR (in SRPROPEN) VRTIO (VR)
PRRD (SR) VROTOPO (in VRTIO)
PRWR (in SRPRRD) VTQ (VR)
PUC (in MRLUC) VTRET (in VLSRET)
PUTCHR (VR) VWRET (VR)
QGDASH (in MRPLTPBK) WALK (in MRPLTPBK)
QVTWR (in VLVPZ) WD (foreground version
RAOBIN (VR) in MRJMBFUN)
RBYTSX (MR) WMIX (VR)
RDTRK (SR) WRBOX (in VLRVA)
READD (MR) Writa (MR)
READDL (in MRREADL) WRTIDU (in MWRTID)
READOF (MR) WRMAR (TR)
RETIO (TR) WRTRK (in SRRDTK)
RORDER (TR) ZECONV (VR)
SATEAR (MR) ZEROS (SR)
SATPOS (MR) ZWIND (in VLGWVA)
SATTV (MR)
SDEST (MR)
SENOUT (foreground version
MRJMBFUN)
SGROAB (MR)
SONDANL (MR)
SOLARP (MR)
SOLUX (NR)
SQLED (MR)
SRCH (TR)
SRGSS (in VLSRVA)
STC (in SRC)
STDAT (TR)
SURGE (TR)
SYMVRT (AR)
TEKPUT (in MRPLTPBK)
TOKANL (MR)
TQ (foreground version
MRJMBFUN)
TQMES (MR)
TQSET (in MRLTO)
TRMNL (MR)
TSNIO (TR)
TVSAT (MR)
ULMR (TR)
UNLOCK (in SRLOCK)
SUBROUTINE DESCRIPTION LIST

1) II (SIZE, VAL, STR, POS): Convert integer to string (fixed length field); 'SIZE' is length of field in bytes, 'VAL' is the integer to convert, 'STR' is the output array, 'POS' is the offset (0-based) at which to start the output. If the integer (plus sign, if necessary) doesn't fit into 'STR', an asterisk is placed in the high order position.

2) ABORT: aborts a program or subroutine (plots 'INIT RESTARTED' on screen).

3) ANGGET: Gets angles (such as local zenith angle (angle between perpendicular to earth at observation point and satellite) and solar zenith angle (same as local zenith angle except sun instead of satellite).

4) ANGLES (JDAY, JTIME, XLAT, XLON, GHA, DEC, SATANG, SUNANG, RELANG): Computes zenith angles to sun and satellite and relative azimuth angle. Input parameters: 'JDAY' = picture day, 'JTIME' = picture start time, 'XLAT' = latitude of point, 'XLON' = longitude of point, 'GHA' = Greenwich hour angle of sun, 'DEC' = declination of sun. Output parameters: 'SATANG' = zenith angle of satellite, 'SUNANG' = zenith angle of sun, 'RELANG' = relative angle.

5) ANGSS (IFLD, ITAG, NCOLS, NROWS, TLAT, WLON, DING, LEV, ICHR, MDNG, NSFC): Functions similarly to SRGSS, except for upper air data rather than surface data, gets guess information at gridpoint locations in gridfile for upper air analysis for whatever parameter is being analyzed (gets data from guess MD file after guess grids have been reformatted into it via program GSVIA).

6) ARASIZ (AREA, LINSIZ, ELESIZ): Creates a file for an image 'LINSIZ' x 'ELESIZ.' Input parameters: 'AREA' = area number, 'LINSIZ' = number of lines requested, 'ELESIZ' = number of elements requested.

7) ATOE (NBYTES, ARRAY, LEFPOS): Convert characters from ASCII to EBCDIC. 'LEFPOS' is 0-based offset within 'ARRAY', 'NBYTES' characters are converted in place. This is one of the character conversion routines.

8) BARB (DIREC, SPEED, XPG, YPG, KOL, SIZE): Simply plots wind barbs; 'DIREC' is direction relative to north (or top of TV screen) from which wind blows, 'SPEED' is in any arbitrary units, 'SIZE' indicates the size of the barb to be determined from experience. Actually, 'SIZE' is the desired length of a 10-knot barb. A negative 'SIZE' denotes southern hemisphere, to place wind barbs on opposite side of arrow.

9) BLKA (N, ARRAY): Set 'N' words of 'ARRAY' to EBCDIC blanks (blanks N words of an array).

10) BOX (DROW, DCOL): Is internal subroutine, does wide line graphics, sets up box.

11) CALDAY (YYDDD, KYEAR, NMON, KDAY, AMON): Converts 'YYDDD' to normal calendar day (year, month, day), month is in three-letterdd form.
12) CLEANA (NBYTES, IARRAY): Change all unprintable characters to blank; 'IARRAY' is array, 'NBYTES' long (clean an array by transforming bad or invalid characters to blanks.

13) CLEANW (NWORDS, IARRAY): Change unprintable characters to blanks; 'IARRAY' is array, 'NWORDS' words ('NWORDS'*4 bytes) long, 'NWORDS' must be less than or equal to 64. Cleans unprintable characters out of 'NWORDS' words, is like subroutine CLEANA, except number of words (not bytes) to be cleaned is specified.

14) CLMGES: Gives the climatological guess for satellite temperature retrievals, output consists of 40 levels (1000-.1 mb) of temperature data, 20 levels (1000-115 mb) of w (mixing ratio) data at 15°, 30°, 45°, 60° and 75°N, from 30°-75°N, have January and July data sets of T and w, 15°N is same data for both January and July, 1000 mb height is accessed within the subroutine proper and outputted through common block SURF, also returns 15-level temperature profile and 6-level dewpoint profile through common block GUESS.

15) CLOSA (IAREA): Closes input area, removes area from lists.

16) CLOSAO (IAREA): Closes an output area (digital segment on disk). One might think of an area as "external memory" (unlike core, which is "internal memory"), also writes onto disc any data in buffer.

17) CO2TAV (PATH): Calculates transmissivities for CO2 channels (is of course channel dependent), output consists of transmittance through each of 40 pressure levels for a given channel.

18) CONTAV (WMIX, PATH): Computes transmittances for water vapor/trace gas continuum.

19) CONTNT (ADDR, LEN, RETURN): Accesses a word or array from memory when its byte length and absolute memory address are known.

20) CSRAOB (IOPT, MD1, MD2, IDAY, ITIME, IDNO, NLEV, ISTDAT, LEV, PRES, TEMP, TD, DIR, SPD, Z, ISTAT): Input raob data from file(s), construct complete sounding for given day/time/station from mandatory and significant levels (get mandatory + significant levels from raob-type files), levels are returned in descending order by pressure, missing data is marked -99999.

**Inputs:** 'IOPT' describes desired options. It is Ø or any sum of:

1: Recompute heights for all levels with p/T/TD data; if this option is not taken, no interpolation of any kind is performed--the raw data only is returned to the caller.

2: Interpolate pressures for the significant level winds.

4: Interpolate missing temperatures and dewpoints.

'MD1' is MD file number of mandatory data file. 'MD2' is MD file number of significant data file. If less than or equal
to $\emptyset$, no significant levels are included. 'IDAY' is day of desired data (YYDDD), 'ITIME' is time of desired data (HH0000), e.g., $\emptyset$ or 120000, 'IDNO' is desired station number.

Outputs: 'NLEVS' is total number of levels returned (maximum of 100). 'ISTDAT' is four word array to contain latitude, longitude, elevation and state code of desired station. (Following arrays should be dimensioned 100. In the real arrays, missing data values are returned as -99999.) 'LEV' is integer array containing the level (for mandatories, "SFC," the integer pressure (mb), "TRO1," "TRO2" or "MAXW," for significant, either "SIGT" or "SIGW"), 'PRES' is real array containing pressure (mb), 'TEMP' is real array containing temperature (deg K), 'TD' is real array containing dewpoint (deg K), 'DIR' is real array containing wind direction, 'SPD' is real array containing wind speed, 'Z' is real array containing height of level, 'ISTAT' is status code ($\emptyset$=OK, but some data may be missing, -1=data missing, no sounding returned, -2=no 'MD1' file).

Note: 'CSRAOB' employs several auxiliary subroutines whose names begin with 'CSRAO-.' Avoid naming other subroutines 'CSRAO-.'

21) CSRAOI (NLEV, PRES, TEMP, TD, DIR, SPD, Z): Routine to interpolate missing values of temperature, dewpoint, wind direction, wind speed. Selected by IOPT in CSRAOB.

22) CSRAOM (LEV, PRES, TEMP, TD, DIR, SPD, Z, NLEVS, JPR): Merge significant levels with mandatories and drop extraneous levels (prefer mandatory to significant data when both present for a given level). 'NLEVS' is number of levels in the various arrays. It is revised (downward) on output. 'JPR' is pointer array giving on input the observations by descending pressure and on output is revised to drop unneeded levels.

23) CSRAOP (NLEVS, JPR, Z, SCALHT, PRES): interpolate pressures for intermediate levels with heights.

24) CSRAOR (NLEVS, JPT, LEV, PRES, TEMP, TD, DIR, SPD, Z, SCALHT): Routine which reorders data arrays and deletes any levels with either missing pressure, temperature or height.

25) CSRAOS (NLEVS, X, JPR): Sort observations on descending 'X,' result is permutation of pointer array 'JPR,' which should be initialized with pointers before entry. Sort into descending-'X' order using pointer array 'JPR.' This is an internal subroutine of CSRAOB.

26) CSRAOZ (NLEVS, JPR, PRES, TEMP, TD, Z, SCALHT): Recompute heights ('Z' array) using $p$, $T$, TD. 'NLEVS' is total number of levels. 'JPR' is a pointer (subscript) array which orders the levels by descending pressure. 'PRES' is the array of pressures. 'TEMP' is the array of temperature. 'TD' is the array of dewpoints. Input/output: the first valid height is selected as a baseline and all subsequent heights are re-computed (whenever $p/T/\text{TD}$ are all present) for hydrostatic consistency. Output: 'SCALHT' is an array of scale heights for the layer just below the current level (will be used later in interpolating pressures).

27) DCLOSE (LUN): Closes an LW (large word array) file.

29) DIRADJ (FLAT, FLONG, SATLON, DIREC, ADJDIR): Makes slight adjustment to any wind to adjust it for parallax when plotting on a grid superimposed over a hemispheric image; in other words, winds will be plotted from their true directions anywhere on the grid.

30) DOPEN (DFILE, LUN, LREC, *): Opens LW (large word array) file. When file is opened, computer is told how many words are to be read; from then on any reads or writes (subroutines DREAD or DWRITE) will access or write out that number of words onto or from THAT FILE (does not apply to other DOPENs and files). Different LW files can have different amounts of words read by different settings in DOPEN.

31) DREAD (LUN, NREC, JBUF, *): Reads from an LW (large word array) file.

32) DSHOFF: Dash off (--).

33) DSHON: Dash on (--).

34) DWRITE (LUN, NREC, JBUF, *): Writes onto an LW (large word array) file.

35) EDEST (CTEXT,IVAL): Display message + integer/A4 on CRT (error message class).

36) EMES (TEXT, VAL): Output error message + integer/A4; 'TEXT' is up to 20 word array of message ending with "$" or "$$." If "$", 'VAL' is treated as integer, if "$$", 'VAL' is treated as integer or A4.

37) ENAREA (AREA, SS, DAY, TIME, LINE, ELEM, INCL, INCE, MEMO): Builds and writes area directory, writes area parameters into area directory. Input parameters: 'AREA' = area number, 'SS' = satellite identification number, 'DAY' = day in YYDDD format, 'TIME' = time in HHMMSS format, 'LINE' = satellite line coordinate of upper lefthand corner, 'ELEM' = satellite element coordinate of upper lefthand corner, 'INCL' = line resolution, 'INCE' = element resolution, 'MEMO' = 8-word integer array to hold a memo on area.

38) ENCODE (FMT, OUTBUF, ARG1, ARG2, ARG3, ...): The encode package for the IBM 4341. It may have as many as 21 or as few as 0 arguments. ENCODE is designed for use with FORTRAN-66, while subroutine ENKODE is to be used with FORTRAN-77. Both ENCODE and ENKODE work in a fashion similar to PRINT statements, with the arguments being the data printed. See DNENCODE for further details.

39) ENCODX (NARGS, LFMT, LBUF, LIST): Special routine which is called only by the ALC routines "ENCODE" and "ENKODE", does the actual encoding of the variables.

40) ENDPLT: Subroutine called at end of a plot, is an interface subroutine to force binding off of buffer. Makes sure all of buffer is plotted on video screen (only applies to video screen), no matter how little information is present.
41) **ENKODE (FMT, OUTBUF, ARG1, ARG2, ARG3, ...):** Used in conjunction with FORTRAN-77, encodes up to 21 variables and places them in a suitable buffer which can then be displayed by a specific display instruction such as LTQ (buffer can also be displayed by suitable construction of the ENKODE statement itself). Variables are encoded into character strings. It is like a formatted "WRITE" statement. The variables can be displayed on screen (CRT) or printer. Arguments: 'FMT' - either an array name or a literal string containing a FORTRAN-type format, describing the manner in which the variables are to be encoded. If it is an array name, it must be type CHARACTER using ENKODE, 'OUTBUF' - an array into which the variables are to be encoded (the array must be sufficiently long to contain the encoded variables, and it must not be type CHARACTER), 'ARGN' the internal variables to be encoded, 21 or fewer, possibly none. See DVENCODE for further details.

42) **ENPT:** An interface subroutine to force binding off of buffer. Makes sure all of buffer is plotted on video screen (only applies to video screen) no matter how little information is present (is called by ENDPLT).

43) **EPOCH (IETIMY, IETIMH, SEMIMA, OECCEN, XMEANA):** Finds time of perigee from Keplerian epoch.

44) **EXTEMP (TT, LAT):** Extrapolates temperature from top of NMC guess (either 50 or 10 mb, depending on guess being used) to 0.1 mb using regression equations based on NMC climatology, EXTEMP used only with NMC guess (either LFM or hemispheric) because climatology guess has 40 levels in it already.

45) **FBARN (XLATN, NR, NC, XGA, XGD, XGE, XDA, XROW, XCOL, NSTA, XINC, IFREQ, IGUESS):** A fast approximation to Barnes interpolation. This routine implements a fast algorithm for a Barnes interpolation of randomly-located observations into a uniform latitude-longitude grid of values. This is done for one set of observations. The computing time of this algorithm is proportional to NR*NC*NSTA*LOG(NSTA), compared to NR*NC*NSTA for INTPO parameters. 'XLATN' is input north latitude of grid top in degrees, 'NR' is input number of rows in grid, 'NC' is input number of columns in grid, 'XGA' is output array (NR, NC) interpolated from 'XDA,' 'XGD' is work array (NR, NC) for weight sum denominator for first pass, 'XGE' is work array (NR, NC) for weight sum denominator for second pass, 'XDA' is input array (NSTA) containing first observed parameter, 'XROW' is input array (NSTA) containing latitudes of observations, 'XCOL' is input array (NSTA) containing longitudes of observations, 'XROW' and 'XCOL' are measured as grid row, column. Thus, the geographical point at the upper left corner of the grid has 'XROW'=1 and 'XCOL'=1. 'NSTA' is input number of observations (stations), 'XINC' is input latitude between rows in degrees, 'IFREQ' is input lowpass filter bandwidth parameter, 'IGUESS' is non-zero if guess grid input through 'XGA' is proportional to spatial frequency. Good values for 'IFREQ' depend on station density per grid square. Too low a value may cause a floating point abort. Values from 30-60 are common on McIDAS. These parameters are identical to the parameters for the INTPO routine in the McIDAS SVCLIB library, except for the addition of the work arrays 'XGD' and 'XGE.' In particular, if 'IFREQ' is the same in this routine and in the routine INTPO, then they should give the same interpolations, up to the approximation. Note that
the values in 'XDA,' 'XROW' and 'XCOL' are changed by this routine, final value is 0 (OK), -1 (bad parameters).

46) FILCLD (NSAM, NRAS, RADS, AVG, STD, BDAT, RMAX, TCLRMN, FAC, NS): Called by subroutine NSTAR, sets up file of clear radiances, but in somewhat different way than FILCLR, is registered (TBB data for given band for given FOV accepted only if all bands (1-10 and 12) have reasonable TBB data; hence, average TBB for any band is always calculated from same FOVs).

47) FILCLR (NSAM, NRAS, RADS, AVG, STD, BOAT, KM, NS): Also called by NSTAR, sets up file of clear radiances, is not registered (TBB data for given band for given FOV accepted even if some other bands for that FOV have unreasonable data; hence, average TBB for different bands may be computed from different FOVs). Both FILCLD and FILCLR filter out noise as well.

48) FILL (IGRID, ITAG, IBUF, IFG, NROW, NCOL): Fills missing elements of grid with appropriate values by linear extrapolation.

49) FILTER (VDAT, NSPIN, ERM, MSAM, TSFS, LDETR): Filters brightness temperatures for "one-step" physical matrix inversion retrieval (done in VLVTP2). Gets retrieval box-averaged brightness temperatures for each band, average surface elevation for retrieval box, expected error of brightness temperature observations, etc.

50) FLOC: Gets latitude and longitude from line and element (for either TOVS or VAS).

51) GAMTA (TAU, KCHAN): Adjusts transmittances for up to 40 levels of a given sounding by raising $\tau$ at each level to an empirical correction factor "G" (for example, $\tau(I) = \tau(I)^G$).

52) GESPRO (ICLIM, NOSFC, MDNO): Get 40 level profile from VASGSS or climatology, add surface. Gets guess profile of temperature and mixing ratio for all 40 retrieval levels (uses guess MD file and statistics).

53) GETDAY: Get system date, in terms of integer values, from computer's own clock (format of YYDDD).

54) GETFRM (FRAME, ENTRY): Reads entry from frame directory "FRAMED," takes input parameter 'FRAME' (frame number) and returns output parameter 'ENTRY' (64-word array containing directory 'ENTRY'). The directory contains such things as frame number, magnification, etc.

55) GETGAM (DAY, TIME, GAMMA, GAMDOT): Gets 'GAMMA,' 'GAMDOT' for 'DAY,' 'TIME' from NAV file. Searches NAV files in order: current file, file 2, file 1. 'GAMMA,' 'GAMDOT' give E-W shift misalignment adjustments—one adjustment per picture.

56) GETNAV (IDAY, IEXIST): Fills GAMCOM, BETCOM with values from NAV (navigation) file, gets navigation information (parameters) from navigation files and puts it in navigation common block. Gets navigation (latitude and longitude) for VISSR images (imaging mode). It is satellite dependent, not wavelength dependent.

57) GETSFV: Gets surface data from gridfile for VAS or TOVS retrievals (1000 mb height, sea level temperature and dewpoint depression).
58) GETTIM: Get system time, returns integer value in HHMMSS format (current time of day) from computers' clock.

59) GRADWI (JHITE, II, JJ, FCORIO, U, V, JHIT2, DELT, JHIT3): Get gradient wind from analyzed field in height, wind direction returned is clockwise from grid north.

60) H2OTAV (PATH): Calculates transmittances for H2O channels.

61) HEAPFY (XDA, XROW, XCOL, LL, K): Sorting routine. Sorts data according to latitude and longitude.


63) HTV (S, LSTA): Calculate heights from 40 level T, W profiles, beginning at station pressure/elevation. File lower mandatory levels by direct linear (ln(p)) interpolation between surface and IZ10.

64) HTX (S, LSTA): Calculates heights from 40-level temperature, moisture profiles beginning at station pressure/elevation. Fills lower mandatory levels by direct linear (ln(p)) interpolation between surface and IZ10 (1000 mb height).

65) IECONV (ITEM, JRMP, LBUF): For use with ENCODE, called by ENCODX, is part of encoding package.

66) IGNAME (GFNO, FILNAM): Construct gridfile name from number. Name is of form "GRIDDNNNN."

67) INITPL (IGD, NW): Entry in PLTPAK to initialize plot routines. 'IGD' = output frame number, or negative of area to use, 'NW' = graphics width. Starts plot package.

68) INTER (XDA, XROW, XCOL, LL, K): Is an entry point within subroutine HEAPFY.

69) INTPTW (PI, TI, WI, PF, TF, WF, NI, NF): Interpolates temperature and mixing ratio in pressure.

70) IQ (MSG): Allows user to conform to FORTRAN-77 keyin standards (MAINO) without having to make significant changes in his/her existing main routine. Brings the values contained within positional parameters into the main program via an array (the argument of the subroutine call).

71) ISQX (CPGM, NVAL, CTOK): Load and start (sequential) execution of McIDAS load module. Control does not return to caller until execution complete. 'CPGM' is load module name (blanks on the right if necessary). In calling program, 'CPGM' is character *8 or more, 'NVAL' is the number of tokens in 'CTOK,' 'CTOK' is an array of character *12 tokens (the first should be the program name and the remaining are parameters). Unless
l-letter keyin (or "CNTRL"), ISQX does load of link. Load/link is mode
via "linker." Transfer vector is passed along with 'NTOK,' 'CTOK.'
Final value is 0 (OK; execution completed), or -1 (couldn't do it).

72) ITOC (SIZE, VAL, STR, POS): Converts an integer to a character string
(variable length of output field). 'SIZE' is maximum size of output
field, in bytes, 'VAL' is integer to be converted (+ or - OK), 'STR' is
destination string (A4), 'POS' is offset (0-based) within string where to
begin output. NOTE: the output takes only as many characters as are
required for all the digits in the string, plus the - if negative. Zero
is output as a single "0." If the number doesn't fit, the high-order
character is set to "." In summation, ITOC converts binary to
EBCDIC-variable width.

73) JMBWTF: Called within subroutine LWSO, does the actual writing of the
block to disk.

74) JSQX (CNAME, CARG1, CARG2, CARG3, ...): Allows a program to perform a
keyin or keyins, rather than the operator or user having to do them. For
instance, a graphic could be plotted by using JSQX rather than manually
punching in keyins. The arguments to the function/subroutine are the
keyins themselves. In other words, JSQX allows one foreground program to
invoke another. The routine is called by either a FORTRAN-66 or 77
program containing the arguments to be passed to Barrett's routine
'ISQX.' As many as 20 CARG*'s may be passed to JSQX. It is unimportant
that the arguments be typed character or otherwise in the calling
routine, because JSQX can determine which they are. The argument 'CNAME'
must contain at least eight characters expressing the keyin name to be
invoked. The remaining arguments contain 12 EBCDIC characters,
expressing the keyed parameters. If the calling program is written in
FORTRAN-77, the arguments need not be 8 or 12 bytes in length. For
example,

    call JSQX ('CNAME,' 'ARG1,' 'ARG2')

is permissible. But in FORTRAN-66, we would need

    call JSQX ('CNAME___,' 'ARG1_______,' 'ARG2________').

The functional value returned by the routine is zero (JSQX was
successful) or non-zero (JSQX was not successful). The routine may also
be called as a subroutine. For example,

    call JSQX (CNAME, CARG1, CARG2, CARG3, ...)

though in this case there is no indication of success or failure.

75) LINFIL (LINE, N, IPASS): Do linear interpolation on values along a
single line.

76) LOCK: Locks a resource with an arbitrary 8-character name, makes sure no
other users can modify particular file being worked on until the work is
done; file is henceforth unlocked by using subroutine UNLOCK.
77) LTQ (LINE): Display a line of 80 characters; 'LINE' is array containing message to display and 'LINE' is displayed on device 0 (black hole), 1 (CRT), 2 (printer) or 3 (system printer) according to setting made by calling subroutine TQSET. If you never call TQSET, output goes to 1 (CRT). Function value is total number of calls made to LTQ so far.

78) LWCL: Close file, remove it from open tables. No harm is done if the file is not already open. Call LWCL (0) causes all files open for this initiator to be closed. It is not strictly necessary to call LWCL (except when writing with locks) as LWCL (0) is called when your program terminates or aborts.

79) LWGET (IFILE, IWORD, NWORDS, IAREA): Service routine for LW (large word array) level disk input/output. Reads 'NWORDS' words from an open McIDAS file, beginning at virtual word 'IWORD,' into 'IAREA.' 'IFILE' is the file name (8 bytes). If value returned (I=LWGET(......)) = 0 (OK), if -1 (request goes beyond the possible file extents). Reading from pages that were never written is OK, the data in the missing pages is returned as HEX 80808080. If the file is not already open for this initiator, it is opened (with an LWOPEN 'IACCESS' = 1). Technical note: the read request is fulfilled through a number of buffers (8 is standard) which are core-resident in the scanner. As a result, consecutive requests for data in a given page, or in a page recently accessed, may not result in actual I-O, but rather in a core-to-core transfer.

80) LWMOOP (CFNAME): Flush LW buffers. Buffers with write-back flags (IBFOMOD value > 0) are dumped. Then all buffers which match 'CFNAME' are set to "EMPTY." All other buffers remain full of data. Exception--if 'CFNAME' = '', all buffers are set to empty. This routine is called automatically at end of each McIDAS program, so it doesn't have to be called explicitly in the program code. This routine replaces old subroutine LWCL (which still works, however).

81) LWNEWF (CFNAME, IPAGE, ISECTR): Make entry into filename tables for new extent. 'CFNAME' is file name, 'IPAGE' is file extent (page number of file's page table) and must be negative, 'ISECTR' is sector location of file's page table, which must already be in proper format and nonempty.

82) LWOPEN (IFILE, IACCESS): Causes a file to be opened. Opens an existing McIDAS file for read access ('IACCESS'=1) or read/write access ('IACCESS'=2). It is permissible to write on a file when it has been opened with 'IACCESS'=1, but it is not safe (loss of data integrity) unless your program is the only program in the system that is allowed to write on the file. If value (I=LWOPEN(......)) returned = 0 (OK), if -1 (no such file). If the open table for your initiator is full (current capacity: 16 files), LWOPEN aborts: U030. Finally, the file opening is recorded in SYSCOM so that the file may be closed automatically when the program terminates.

83) LWPO (ICB): Write a file page out to disk. 'ICB' is the number of a buffer containing the file page.

84) LWSO (ISECTR, IBUF): Writes block to disk (page), called by LWI and LWO, is part of McIDAS operating system.
85) MDCLOS (MDNO): Close MD file(s). 'MDNO' is MD number of file to be closed, or 0 to close all.

86) MDNAME (MDNA, FILNAM): Make name of MD file.

87) MNRAOB (MD, IDAY, ITIME, IDNO, ISTDAT, LEVDAT, ISTAT): Gets mandatory level raob data. 'MD' is MD file number, 'IDAY' (YYDDD) and 'ITIME' (HH0000) define the day/time, 'IDNO' is the station number, 'ISTDAT' is returned as a 4-word array: LAT (degrees *10000), LON (degrees *10000), ZO(surface elevation in meters), STATE ID (A4 format). If any of these are not in the schema, Z80808080 is returned. The level data is returned in array 'LEVDAT' (dimensioned 7*30); that is, maximum of 16 levels returned. Actual number will be in 'ISTAT'. Level (A4 format) is same as p or "SFC" or "TRO." P is in mb*10, T (K*100), TD (K*100), DIR (wind direction in degrees), SPD (wind speed in meters per second), and Z (in meters). 'ISTAT' is number of levels (OK), -1 (data not present), or -2 (invalid MD file). MNRAOB will extract data from any MD file whose schema has the following properties: (1) keys 'day,' 'time' are present and in the row header; (2) key 'ID' identifying the station is present in the column header or in the data portion, (3) the data portion contains from 1 to 16 repeat groups, each with 7 fields in the order: LEV, P, T, TD, DIR, SPD, Z. Note that the schema need not contain column headers.

88) MOVB (NBYTES, FROM, TO, TOPOS): Moves N bytes from word boundary to anywhere. 'TOPOS' is offset within 'TO' array (0-based), if 'NBYTES' is less than or equal to 0, nothing happens.

89) MOVF (NBYTES, FROM, FRMPOS, TO, TOPOS): Move N characters with source/destination offsets. 'FRMPOS' is offset within 'FROM' array (0-based), 'TOPOS' is offset within 'TO' array (0-based). If 'NBYTES'/NWORDS is less than or equal to 0, nothing happens. MOVF essentially moves N bytes from anywhere to anywhere.

90) MOVCW (CARR, N, IARR): Gets literal data into non-character-type variables, move character variable to word array; 'CARR,' 'N' represents character variable of length 'N,' 'IARR' is destination array.

91) MOVW (NWORDS, FROM, TO): Moves N words from one array to another. If 'NWORDS' is less than or equal to 0, nothing happens.

92) MOVWC (IARR, CARR, N): Move word data to character variable. 'IARR' is source array, 'CARR,' 'N' represents character variable of length 'N.'

93) NSTAR (RADS, VCLR, IFAIL): Infers clear column radiances from partially-clouded fields of view.

94) NVINIT (BETAIN, BETDOT, INAV, PTIME): Refers to opening up of navigation (NAV) file.

95) O3TAV (TOTO, PATH): Computes transmittances for channels affected by ozone absorption (especially channel 2 (approximately 690 cm⁻¹)).

96) OPNA (IAREA): Collect a list of the track groups assigned to the area and place them in the common block.
97) OUTINT (KOUT): Outputs integer values (maximum of 9) in an array onto the video screen (CRT) or printer. It is somewhat tedious, since only one array at a time can be printed, whereas subroutines ENKODE/ENCODE can print several variables (including arrays) at once. In addition, OUTINT works only for integers, while ENKODE/ENCODE can print characters and reals, in addition to integers.

98) PACK (N, IS, ID): Moves 'N' bytes from 'IS' at one per word to 'ID' at four per word.

99) PAGE (LINB, NELB, LINE, NELE, N): Basic graphics package (WRRRM only, WRRRM refers to old graphics program). Modified to compute distances to edge of CRT, defines plotting area.

100) PENADD (IVAL): WRRRM ("Write Random Read Raster Memory") interface subroutine to add 3-bit values.

101) PENBEG: New WRRRM interface subroutine to begin a new packet, deals with terminal protocol when new line is to be added.

102) PENMOV (ILINED, IPIXLD): WRRRM ("Write Random Read Raster Memory") interface subroutine to move the pen a "delta."

103) PLNKIV (ISAT): Brings in coefficients used for any Planck function calculations from disk.

104) PLOT (LIN, NELEM, IPN): Is called to plot a line. It does any scaling requested, checks that the points will go on the screen and adjusts the line if not. In addition, PLOT handles the generation of wide and dashed lines. Note that the last point of the line is not displayed under the assumption that it will be the first point displayed on the next call. 'LIN' and 'NELEM' represent the logical end of a line.

105) PLTDIG (LINE, LEL, NVAL, LHGT, NFW, LEVEL): Plots a single digit.

106) POST: Part of PRINT package, "wakes up" printer handlers. Is valid only in McIDAS.

107) PRCL: Is an entry within PRRD on the line immediately below the PRWR entry. Closes a transfer file.

108) PRCLOS (I): End a print spool string and connect it to que for printer 'I.'

109) PREATV (TEMP, WMIX): Takes atmospheric information, gets quantities required for transmittance model (also computes derived functions of temperature and mixing ratio).

110) PRECW (P, W, U, NP): Calculates precipitable water for a given atmospheric layer or layers using pressure and mixing ratio. Calculates 'U' between two given p-levels.
111) PRETAV (ISAT, KCHAN): More preparation for transmittance calculations, initializes pressure functions and reads in coefficients.

112) PRIOUT (JDEV): To regulate printed output, as follows:

   0. Suppress all LTQ-generated output (black hole).

   1. To terminal monitor

   2. Local printer if there is one, otherwise to second floor, provided you are at a second floor terminal.


113) PRLINK (N, LINE): Spooled printing subroutine. Prints a line on printer number 'N' (≥ 0); 'N' = 0 implies system printer. Interleaving lines to different printers is possible, but each time the printer number is changed, the last batch of lines is tied off and spooled out. Hence, the printer number should be changed only infrequently. To cause a skip to top of page, call PRLINK with LINE (1)=-1. Calling PRLINK with 'N' < 0 just closes any open printer.

114) PROFIX (PR, TR, DR, WR, P, T, W, IL, IM, IS, IT, LAT): Takes first guess profile from NMC grid and transforms it to a form needed for internal levels (temperature at 15 levels becomes temperature at 40 levels; dewpoint at 6 levels becomes mixing ratio at 15 levels, etc.).

115) PROPEN: Start a print spool string.

116) PRPRPR (K): Add a line of print to spool.

117) PRRD: Printer spool read/write, reads record from transfer file, is not called by user.

118) PRWR: Is also an entry within PRRD, writes record on transfer file, is not called by user.

119) PUC (FROM, INDEX): "Put user common," pokes word into user common. Subroutines LUC/PUC map user-common indices into indices in core-resident SYSCOM. The constants required for this mapping are computed when the system (the keyin scanner) calls "INILUC." System must call "INILUC" before LUC or PUC can be called. 'INDEX' is where (in user common) to do transfer, 'FROM' is fullword data value to transfer to user common, final value is word transferred.

120) PUTCHR (NBYTE, ARRAY, ITEM): Stores simple byte in an array (the opposite of LCHR).

121) QGDASH (JDASH): Return current status of dash mode.


123) RAOBIN (P, T, W, IOK): Ingest radiosonde data, gets radiosonde report (mandatory + significant level data).
124) RBYTSX (IAREA, NREC, IOFF, NUMBYT, IARRAY): Read a line from an area, returning a specified subset of the line. 'IAREA' = area number, 'NREC' = line offset (0-based), 'IOFF' = offset to first byte desired. 0 returns the first byte of the data. Negative offsets are used to access the prefix. A large negative offset will reference the first byte of the prefix. 'NUMBYT' = number of bytes desired; will not return any past the line end. 'IARRAY' = (output) array to hold returned data.

125) RDTRK (INDEX, TRACK, ARRAY): Area access method BDAM read/write, reads a track off an area file. 'INDEX' is a number between 0 and 7 specifying which DCB is used, 'TRACK' is a number between 0 and 16383 specifying track number, 'ARRAY' is the I O buffer address.

126) READD (ANUM, ENTRY): Reads directory entry from area/sounding directory "DATDIR." Input parameter: 'ANUM' = area number. Output parameter: 'ENTRY' = 64 word directory entry (integer array).

127) READDL (ANUM, ENTRY): Locks and reads directory entry from area/sounding directory "DATDIR." Input parameter: 'ANUM' = area number. Output parameter: 'ENTRY' = 64 word directory entry (integer array).

128) READOF (IAREA, LINOFF, IOFF, NUMBYT, IARRAY): Read a line from an area, returning a specified subset of the line. 'IAREA' = area number, 'LINOFF' = line offset (0-based), 'IOFF' = offset to first byte desired. 0 returns the first byte of the data. Negative offsets are used to access the prefix. A large negative offset will reference the first byte of the prefix. 'NUMBYT' = number of bytes desired. Will not return any past the line end. 'IARRAY' = (output) array to hold returned data.

129) RETIO (IOUT, NSEC, IOPT): TOVS equivalent to VRTIO, except doesn't access any context file or MD file. Only accesses TOVS retrieval files.

130) RORDER (B, A, NI, M, IDIR): Orders the data in the array 'B' from smallest to largest ('IDIR'=1), or largest to smallest ('IDIR'=2). The reordered data is returned in the array 'A.' Upon completion of the ordering, the array 'NI' contains the original positional subscripts of 'B' in the new order. This index array can then be used to order other data associated with the data in 'B.' 'M' is the number of elements to be ordered.

131) SATEAR (PICTIM, XLIN, XELE, XLAT, XLON, ITYPE, INAV, BETAIN, BETDOT, ATFRAC): Most general satellite-earth coordinate transformation program, computes satellite, earth coordinates, earth edges, sub points. T(0) is defined to be Greenwich hour 0 of day of navigation. Latitude ranges from +90 north to -90 south. Longitude ranges from +180 east to -180 west. Input parameters: 'PICTIM' = picture start time (hours from T(0)), 'XLIN' = satellite coordinate (line), 'XELE' = satellite coordinate (element), 'XLAT' = earth coordinate (degrees latitude), 'XLON' = earth coordinate (degrees longitude), 'ITYPE' = 1 for satellite coordinate to earth coordinate transformation, 2 for earth coordinate to satellite coordinate transformation, etc. (see MRSATEAR for more information). 'INAV' = +1 - for use of updated
navigation parameters (sets limit to 5), 0 for use of previous navigation
parameters, -1 for use of updated navigation parameters (sets limit to
2), 'BETAIN' = beta angle at T(0) (elements), 'BETDOT' = rate of change
of beta (elements per hour), 'ATFRAC' = cloud height coefficient (ranges
from 0 to 1).

132) SATPOS (INORB, NTIME, X, Y, Z): Generates satellite position vector from
earth center. Input parameters (all integers): 'INORB' = initialization
flag (should = 0 on first call to SATPOS, 1 on all subsequent calls),
'NTIME' = time (hours, minutes, seconds) in HHMMSS format. Output
parameters (all floating point): 'X', 'Y', 'Z' = coordinates of position
vector.

133) SATTV (FRAM, ILIN, ILEL, ITVLIN, ITVELE, SS, JDAY, TIME): Transforms
from satellite to TV coordinates. Input parameters: 'FRAM' = frame
number, 'ILIN' = line satellite coordinate, 'ILEL' = element satellite
coordinate. Output parameters: 'ITVLIN' = line TV coordinate, 'ITVELE'
= element TV coordinate, 'SS' = satellite identification number, 'JDAY' =
YYDDD of frame, 'TIME' = HHMMSS of frame.

134) SDEST (CTEXT, IVAL): Display 'CTEXT' + integer/A4 on CRT (standard
message class), displays message at standard message destination (similar
to subroutine TQMES).

135) SENOUT: Changes terminal state, sends arbitrary messages to terminal.

136) SGRAOB (MD, IDAY, ITIME, IDNO, NSIGT, NSIGW, ISND): Returns (1 station
per call) significant level raob reports. Input parameters: 'MD' is MD
file number, 'IDAY' (YYDDD) is day, 'ITIME' (HH0000) is nominal time,
usually 0 or 120000, 'IDNO' is station number. Output parameters:
'NSIGT' is number of returned significant temperatures (maximum is 100).
If less than 0, MD file is bad or not schema "RSIG," 'NSIGW' is number of
returned significant winds (maximum is 100), 'ISND' is array (7*100) to
contain significant temperatures and winds. Each column contains level
(either "SIGT" or "SIGW"), p is in mb *10, T(K*10), TD (K*10), DIR
(degrees), SPD (meters per second), Z (meters). Note that for "SIGT,"
DIR, SPD and Z will be missing (=Z80808080), and for "SIGW," P, T and W
will be missing. Function value is 0 (OK), or -1 (bad MD file
specifications). Note: This subroutine is intended only for files with
schema RSIG and assumes the positions of variables within the schema.
If schema RSIG is changed, data values for LOCWDDAY and LOCREP may have
to change.

137) SNDANL (HOUR, NLEV, P, T, TD, DIR, SPD, STABIL): Subroutine to compute
stability indices for given sounding. Such as total-totals, lifted, K
and SWEAT. 'HOUR' = HH.00 of raob time (used for parcel characteristics),
'NLEV' = number of input levels in data arrays, 'P' = pressure in mb, 'T'
= temperature in degrees Kelvin, 'TD' = dewpoint temperature (K), 'DIR' =
wind direction, in degrees, 'SPD' = wind speed, in meters per second,
'STABIL' = array containing outputted stability indices.
138) SOLARP (JDAY, JTIME, GHA, DEC, XLAT, XLOI): Computes Greenwich hour angle and declination of sun. **Input parameters:** 'JDAY' = satellite year/day, 'JTIME' = hour/minute/second. **Output parameters:** 'GHA' Greenwich hour angle, 'DEC' = declination, 'XLAT' = latitude of sun position, 'XLOI' = longitude of sun position.

139) SOLVEX (GAM, X, XIT, XTX, XIV, NF, NE): Calculates \((\phi^T\phi)^{-1}\phi^T\). See Smith and Woolf, 1984, equation (6) therein.

140) SQSLED (ICODE): Set line (LTQ) device. Error (EMES) device, debug (DMES) device into UC (user common). 'ICODE' is 1-3 letters (Al, A2 or A3 format) which specify the output devices for the 3 message categories. The first letter directs the "line output" category, the second the "error message" category and the third the "debug message" category. The letters can be: C . . . CRT, P . . . local printer, S . . . system printer, or N . . . nowhere (output disappears). The values designated are recorded in UC words 31-33.

141) SRCH (LATS, LONGS, IM, ILIN, IELE, NPTS, NROWS): A subroutine which uses trigonometry to locate latitude/longitude coordinates in terms of TIROS-N line and element number. Basic framework is determined from top and bottom line numbers fed by 'NROWS' and LTOP. 'LONGS' is +east, -east.

142) SRGSS (IFLD, IDL, NCOLS, NROWS, TLAT, WLON, DINC, IP, MDNG): Gets guess information from guess MD file to use in creating surface analysis grids (for VAS or TOVS). Gets guess information at gridpoint locations.

143) STG (VAL, STR, J): Store a character at offset 'J' in string, store character in rightmost byte of 'VAL' at offset 'J' in 'STR' ('J' is 0-based), final value is 'VAL'.


145) SURGES (NOSFC): Get values from gridpoint surface analysis for TOVS or VAS, according to latitude and longitude (output is 1000 mb height, surface temperature, surface dewpoint depression).


147) TEKPUT (L, P): In graphics package, outputs to Techtronics (4010) compatible displays.

148) TOKANL (CPARR, NTK, NKW, NARR, IDEVAL): Analyze token sequence produced by MCTKNO. 'CPARR' contains the token sequence, 'NTK' is total number of tokens. **Outputs:** 'NKW' is the number of keywords found, which equal the number of items stored in 'NARR'. 'NARR' includes one item for the positional tokens and one for the field (if any). 'NARR' contains one entry for each keyword which gives the number of tokens belonging to that keyword (includes the keyword name), 'IDEVAL' (integer) contains the first four characters of the value assigned to keyword DEV= . (If DEV= not present, 'IDEVAL' contains blanks.)
149) TQ (BUFFER): Involved with writing message to terminal, is part of operating system, sends a text message to CRT.

150) TQMES (TEXT, VAL): Output 'TEXT' + integer/A4 on CRT. 'TEXT' is up to 20 word array of MSG, ending with "$" or "$$" if "$", 'VAL' treated as integer, if "$$$" 'VAL' treated as integer or A4.

151) TQSET (DVC): Redirects destination of standard class error messages. Set/examine display device. To set LTQ device, 'DVC' should be 0 (black hole, no output), 1 (CRT), 2 (local printer), 3 (system printer). To see what is current LTQ device, 'DVC' should be -1; current device is initially set to value in UC (-31); function value is current LTQ device.

152) TRNML (IS): Returns the terminal number.

153) TSNIO (IOP, IFL, IR, IC, NR, NC, IA): TIROS-N data input/output (operates on orbit/image file).

154) TVSAT (FRAM, ITVLIN, ITVELE, ILIN, IELE, SS, JDAY, TIME): Transforms from TV to satellite coordinates. Output parameters give line and element satellite coordinates (Note: satellite coordinates are often called simply "line" and "element") corresponding to input TV coordinates, as well as satellite identification number, date of frame image in YYDDD, plus time of frame image in HHMMSS.

155) ULMR (W): Upper level mixing ratio. Extrapolates moisture from 300 mb up to 70 mb, thereby giving a smooth vertical decrease in moisture rather than a sudden discontinuity (W decreases ultimately to .001 g/Kg for the top 20 levels of the standard atmosphere, so after guess is complete moisture will be .001 g/Kg from 70 to 0.1 mb).

156) UNLOK: Releases file for subsequent use by other programs.

157) VALUE (LAN, LAS, LOW, LOE, INC, IDL, FSCL, VAL): Given latitude and longitude, returns a value interpolated from a grid.

158) VASDAT (IL, IE, VDAT): Routine to access VAS sounder file according to image line and element obtained from subroutine TVSAT, if VDAT(1)=-1, returns only navigation data, if JDAY=0, navigation skipped. 'IL', 'IE' are cursor line and element positions, 'VDAT' is array used to return brightness temperatures at cursor location for different VAS bands. Note: IL and IE are changed in this routine.

159) VASDIFG (IRAS, IPIC, ITEM, MAC, IWD, KOLOR): Simply calls subroutines INITPL, PLTDIG and ENDPLT, interfaces to plotting routines, causes a quantity to be plotted on video screen.

160) VASGES (MDNO, FLAT, FLON): Enters and extracts guess vectors (arrays of T or TD for 21 or 40 levels) of temperature, dewpoint from the VASGSS file (created by program GSVA). Guess values are moved through common blocks /GUES/ and /SURF/.

161) VASNAV (FNUM, SYDREC, ISYD, IHMS): Partially sets up navigation common block, is VAS navigation package (sounding mode).
162) VASRTE (TAU, TEMP, TSPC, RAD, DBDT, TBB, DBDTBB, KCHAN, LCND): Calculates radiance being emitted toward the VAS via the radiative transfer equation (RTE). Calculates total radiance (surface + atmospheric), integrating downward from 0.1 mb to the surface, also gives profiles of \( \frac{\partial B}{\partial T} (\gamma, T^n) \) where \( T^n \) = atmospheric temperature (this is one term used in VAS weighting function), as well as \( \frac{\partial B}{\partial T_B}(\gamma, T^n) \), where \( T_B^n \) = brightness temperature for a given channel calculated from the RTE and the Planck function. Note: VAS weighting function:

\[
W(p) = \frac{\partial T}{\partial \ln p} (\gamma, p, \theta) \cdot \frac{\partial B}{\partial T} (\gamma, T^n) \bigg|_p / \frac{\partial B}{\partial T_B} (\gamma, T^n) \bigg|_T
\]

163) VASTAU (TEMP, WMIX, TOTO, ZENANG, TAU, ISAT, KCHAN): Subroutine which calculates atmospheric transmittances for VAS channels through each of 40 pressure levels for a given channel. Transmittance will vary depending upon which channel is being used, as well as atmospheric transmittances at that channel's wavelength for CO2, water vapor/trace gas continuum, H2O vapor and ozone.

164) VASTPW (TBB, EBB, TOTO, TSFS, URET, ITMAX, ISAT, KCT, NP): Retrieves total precipitable water from VAS radiances.

165) VRTIO (IREC, NREC, IOPT): VAS retrieval input/output. Data from MD file and to MD file. Accesses VASTEXT (context file) which gives context in which program is running.

166) VRTOPO (MLAT, MLON, IELEV, ICHAR, NIN, IBEG): Functions like HRTOP0, except is optimized for VAS usage. Input parameters: 'MLAT' (Ø=89, +N, -S, *100), 'MLON' (Ø=180, +E, -W, *100). Output parameters: 'IELEV' (elevation in meters), 'ICHAR' (surface characteristic, Ø=ocean, 1=land).

167) VTQ (TQBUF): Same as TQ on the Harris, fills array 'TQBUF'(20) with 20 values of 4H. Works the same as LTQ, except buffer is cleared as well (is not cleared in LTQ). Displays contents of subroutine ENKODE buffer.

168) VTRRET (KCl, KC2, DELB, KUSE, WG, IS, M, ER, IOK): VAS temperature retrieval enhancement routine.

169) VWRET (TS, TAUW, U, NL, NLS, TSP, USP): VAS water vapor retrieval enhancement routine.

170) WALK (DROW, DCOL): Is internal subroutine related to wide line graphics, does actual plotting.

171) WD: Retrieves words from user common (UC), is part of the scanner (operating system), alters SYSCOM (systems communications region).

172) WMIX (P, T, DD, W, NL): Calculates mixing ratios for 'NL' atmospheric levels using pressure 'P', temperature 'T', dewpoint depression 'DD' and saturation vapor pressure (ES) at the dewpoint. Note: saturation vapor pressure at the dewpoint = vapor pressure at the dewpoint.
173) WRBOX (IR, IP, IZR, IZP, IC): Writes (plots) a box in color 'IC.'

174) WRITA (IAREA, NREC, IADDR): Writes into an area.

175) WRITDU (ANUM, ENTRY): Writes and unlocks area/sounding directory. Input parameters: 'ANUM' = area number, 'ENTRY' = 64-word directory entry to be written (integer array).

176) WRRMAR (IR, IP, IZ, IC, IN): Program to operate "worm" (WRRRM-"Write Random Read Raster Memory"--graphics package) and indicate on graphics frame where retrieval has been attempted or deleted. 'IL' is line, 'IE' is element, 'IZ' is size of indicator (all in TV units). 'IN' is 1 for square, 0 for "X." 'IC' is color: 1-red, 2-green, 3-yellow.

177) WRTRK: Writes onto a track, is an entry within SRRDTRK.

178) ZECNV (ITEM, NBYTES, JRMP, OUTBUF): Used by subroutine ENCODE, is for conversion from integer to sexadecimal.

179) ZEROS: Sends a bunch of zeroes to FORTRAN, makes all elements of an array equal to zero.

180) ZWIND (JHIT, I, J, JLAT, SPEED, DIREC, JHIT2, DELT, JHIT3): Calculates a given type of wind (geostrophic, isallobaric, ageostrophic) according to the input arguments, also indirectly calculates gradient wind by calling subroutine GRADWI.
APPENDIX IV

Function List and Descriptions

Appendix IV contains descriptions of functions called within VAS retrieval software. However, the only functions described here are functions which are called in the main programs themselves, not the functions called within subroutines subservient to the main programs. In this sense, Appendix IV cannot be considered as comprehensive as Appendix III.

As in the third appendix, the McIDAS source library location for each function is listed, as well as the argument/parameter list and descriptions of the argument(s)/parameter(s) for each function whenever possible. Finally, the purpose of each function is also detailed.
### COMPLETE ALPHABETICAL FUNCTION LIST

1. **AZMUTH** *(VR)*
2. **CHOP** *(SR)*
3. **CKWP** *(MR)*
4. **CLIT** *(SR)*
5. **CPP** *(MR)*
6. **DEWPT** *(TR)*
7. **DLIT** *(ML)*
8. **DKWP** *(MR)*
9. **DPP** *(MR)*
10. **ICURG** *(in VLXRA)*
11. **IGGET** *(in MRIGMAKE)*
12. **IGOPEN** *(in MRIGMAKE)*
13. **IGPUT** *(in MRIGMAKE)*
14. **IKWP** *(MR)*
15. **ILALO** *(MR)*
16. **IPP** *(MR)*
17. **IROUND** *(MR)*
18. **ISATNV** *(VR)*
19. **IVASCL** *(VR)*
20. **LANSEA** *(TR)*
21. **LIT** *(SR)*
22. **LOGAND** *(in VRLOGOR)*
23. **LUC** *(MR)*
24. **LWL** *(in MRLWSUBS)*
25. **MDGET** *(in MRMDMAKE)*
26. **MDINFO** *(MR)*
27. **MDKEYS** *(MR)*
28. **MDOPEN** *(in MRMDMAKE)*
29. **MDPUT** *(in MRMDMAKE)*
30. **RADENC** *(VR)*
31. **VBRITE** *(in subroutine PLNKIV)*
32. **VBDTAU** *(VR)*
33. **VBDDBT** *(in subroutine PLNKIV)*
34. **VPLANC** *(in subroutine PLNKIV)*
35. **VSINT** *(VR)*
36. **WSAT** *(VR)*
FUNCTION DESCRIPTION LIST

1) AZMUTH (ALAT, ALON, BLAT, BLON, DIST): To get azimuth from point A to point B with distance between returned as a by-product.

2) CHOP (X, YMIN, YMAX): Limit size of 'X' to values between and including 'YMIN' and 'YMAX.'

3) CKWP (CKW, I, CDEFLT): Get a program keyword parameter in character string form. 'CKW' is character and is "POS" or the name of a keyword, 'I' specifies which argument from the argument sequence to take (1-based). For example, "POS," 1 specifies the first positional parameter, 'I'=Ø gives the program name for positional parameters ('CKW'=""), or "="/CKW for keyword parameters (if present). 'CDEFLT' is character * 12 to use as missing value. Function value is character string representing the desired parameter or 'CDEFLT' if parameter is missing.

4) CLIT (I): Type transformer: integer/real to character * 4.

5) CPP (I, CDEFLT): Get positional parameter (character * 12). Function value is 'I'\textsuperscript{th} positional parameter or 'CDEFLT' if parameter is missing.


7) DLIT (C): Returns own argument (literal). Result is real * 8 bitwise identical to 'C,' 'C' is character * 8.

8) DKW (CKW, I, DDEFLT): Get keyword parameter (real*8), returns real*8 value.

9) DPP (I, DDEFLT): Get a positional parameter (real * 8).

10) ICURG (IL, IE, IRINC, IPINC): Get response from user, cull out TV line and element as well as cursor dimensions. Function value of 1 if user says 'END,' else Ø. Output parameters: 'IL,' 'IE' are cursor line/element positions.

11) IGGET (GFNO, GNO, MAXWDS, GRID, NR, NC, TABLE): Get a grid from a gridfile. 'GFNO' is gridfile number, 'GNO' is grid number within gridfile (1...999), 'MAXWDS' is maximum size of grid allowed to read, 'GRID' is array to contain grid, 'NR' is returned as number of rows in grid, 'NC' is returned as number of columns in grid, 'TABLE' is 64-word array to receive grid header. Final value is Ø (OK), -1 (no such grid or too big), or -2 (no such file).

12) IGOPEN (GFNO, FILNAM): Open gridfile, return file reference number. 'GFNO' is grid file number, 'FILNAM' is returned as file name of the specified gridfile (8-characters). Final value is Ø (OK), or -1 (can't, e.g., no such file or not a gridfile).

13) IGPUT (GFNO, IGN0, GRID, NR, NC, TABLE, GNO): Put a grid into a gridfile. 'GFNO' is gridfile number, 'IGN0:' if + or Ø, grid is written
in next empty slot after 'IGNO,' if negative, grid is written in ABS ('IGNO'), overwriting any grid that is there. 'GRID' is the grid array, 'NR' is number of rows in grid, 'NC' is number of columns in grid. If 'NR' or 'NC' is less than or equal to Ø, grid 'IGNO' is deleted from file. 'TABLE' is 64-word grid header. Caller must set it up, except that IGPUT stuffs in the 'NR' and 'NC' fields, 'IGNO' is returned as actual grid number stored, final value is Ø (OK), -1 (no room) or -2 (no such grid file).

14) IKWP (CKW, I, IDEFLT): Get keyword parameter (KP) (integer). The keyword (first argument) is searched for in program keyin. If found, the first, second, third, etc., integer value denoted by the second argument is given to the variable. If the KP is not defined, then the variable is given the third argument's integer value by default. For example, the statement I=IKWP('IAD',2,Ø) causes a search for the keyword parameter 'IAD.' If it is found, the second value of the keyword parameter is assigned to variable I. Otherwise, if 'IAD' is not found, I is assigned the value Ø by default.

15) ILALO (X): Changes a floating point latitude-longitude to a packed integer (sign DDD.MM,SS). Input parameter: 'X'=floating point latitude or longitude.

16) IPP (I, IDEFLT): Get positional parameter (integer). The first argument tells computer which value of the positional parameter to look for and assign to the variable. The second argument is the default value assigned to the variable if no such value of the positional parameter has been keyed in (or if the positional parameter has not been keyed in at all).

17) IROUND (X): Rounds a floating point value. Input parameters: 'X'=floating point value.

18) ISATNV (JSAT): Obtain VAS satellite number from 'JSAT' (obtained via TVSAT).

19) IVASCCL (TBO, FAC): Predicts VAS brightness temperatures at 2210 cm⁻¹ based on brightness temperatures at 715, 750 and 790 cm⁻¹; compare with observed and return the following: Ø if ABS (PRE-OBS) is within 'FAC' standard error, 1 if ABS (PRE-OBS) is not within 'FAC' standard error, 2 if check cannot be done because of missing data.

20) LANSEA (ILAT, ILON, LEVEL): Distinguishes between land or ocean, using a low resolution topography (1 degree resolution in both latitude and longitude, as opposed to HRTopo, which uses 10 minute resolution). Is used when low accuracy distinction is needed. If result shows land, then elevation is also returned. One enters with latitude/longitude (positive E, negative W) in degrees * 10⁶. Final value=2 (ocean) or 1 (land).

21) LIT (C): Returns integer bitwise identical to character * 4. This is type transformer for character * 4. 'C' is character * 4 datum (character * 8 for DLIT). Result of (LIT, ALIT, DLIT) is (integer, real * 4, real * 8) bitwise identical to 'C.'

22) LOGAND (A, B): Logical 'and' of two 4-byte arguments.
23) LUC (INDEX): "Look at user common," subroutines LUC/PUC map user-common indices into indices in core-resident SYSCOM. The constants required for this mapping are computed when the system (the keyin scanner) calls "INILUC." System must call "INILUC" before LUC or PUC can be called. 'INDEX' is where (in user common) to do transfer. If greater than $\phi$, terminal-related area is addressed, if less than $\phi$, initiator-related area is addressed. Final value is word extracted from user common.

24) LWI (CFNAME, IBEGWD, NWDS, IARR): This is the service routine for LW (large word)—level disk input-output, entry points LWI (input), LWO (output). 'CFNAME' is file name (12 bytes), 'IBEGWD' is first virtual word in transfer ($\phi$ based), 'NWDS' is number of words to transfer (if 'NWDS' less than or equal to $\phi$, nothing happens), 'IARR' is an array containing record to read/write. Final value for LWI: if $\phi$ (last data transferred was from an actual (allocated) file page), or -1 (last data transferred was from a null page, and was equal to Z80808080). Note: the function value from LWO should be ignored (always $\phi$).

25) MDGET (MDNO, M, N, ARRAY): Get record from MD file. 'MDNO' is the MD file number, 'M', 'N' are the row and column numbers of the desired record. If 'M'=$\phi$, a column header is designated. If 'N'=$\phi$, a row header is designated. 'ARRAY' must be long enough to contain the desired record—all of it, not just the portion being read. This is because row headers, column headers and data records are each read into different regions within 'ARRAY,' so that they will match up properly with their respective keys. If final value=$\phi$ (OK), if=-1 (can't, i.e., no such record).

26) MDINFO (MDNO, MDHD): Check existence of MD file, return header. 'MDNO' is MD file number, 'MDHD' is 64-word array to receive MD file header. Function value is $\phi$ (OK) or -1 (can't, file doesn't exist). Note: last seven words of 'MDHD' are initialized by MDINFO for buffer management. See MDI (MRMDI).

27) MDKEYS (MDNO, NL1ST, LIST, SCALES, UNITS, LOC1S): Returns information corresponding to keys in schema. Returns meteorological data (MD) values/units/location of keys in MD file. 'MDNO' is the MD file number, 'NL1ST' is -1 to retrieve all keys in schema or else the number of keys in the list, 'LIST' is an array of keys, 1 per word (unless 'NL1ST'=-1), SCALES' is an array of words to receive scale factors, 'UNITS' is an array of words to receive key units (A4), 'LOC1S' is an array of words to receive subscripts within a complete (row header + column header + data record) MD file record where the fields specified by the keys are to be found. (Each such field is exactly one word long.) Final value is number of keys for which information is returned. If 'NL1ST' is greater than $\phi$. Thus final value is less than or equal to 'NL1ST'; if not equal, it indicates that one or more keys were not found in schema. If 'NL1ST'=-1, final value is total number of keys in schema (less than or equal to 4$\phi$); final value is -1 if error (i.e., file does not exist). When 'NL1ST' is -1, arrays should be at least 4$\phi$ words in order to contain maximum possible schema. Since the MD file structure permits multiple fields with the same name, MDKEYS is careful to return the values for the first key in the schema for the first occurrence of that key in 'LIST,' the values for the second key in the schema for the second occurrence, and so on.
28) MDOPEN (MDNO, ACCESS): Open MD file. 'MDNO' is MD file number (1-9999), 'ACCESS' is 1 (open for read) or 2 (open for read/write). Final value is 0 (OK or already open) or -1 (can't open).

29) MDPUT (MDNO, M, N, ARRAY): Put record to MD file. 'MDNO' is MD file number 'M', 'N' are row, column number of desired record. If M=0, a row header must be designated. If n=0, a column header must be designated. 'ARRAY' must be long enough to contain the desired record, all of it, not just the portion being read. This is done because row headers, column headers and data records are each read into different regions within 'ARRAY' so that they will match up properly with their respective keys. If final value=0 (OK), if it =-1 (can't, i.e., no such record).

30) RADENC (VALUE, CHANL, DELTAF, YSUBZ): Converts radiance in given channel from counts to radiance in units of mWm⁻² str⁻¹ cm⁻¹.

31) VBRITE (R, K): Calculates brightness temperature based on given radiance and wavenumber.

32) VBDTAU (TAU, B, BS, KCHAN, NL): Calculates radiance using radiative transfer equation (RTE--integrated from top to bottom of atmosphere at retrieval location, i.e., integrated from top down through last full layer above ground). Final radiance includes both surface and atmospheric contributions as well as an empirical correction to the radiance based on the channel being used. Radiance returned is at least 0.001 mWm⁻² str⁻¹ cm⁻¹.

33) VDBDTC (TBB, K): Computes derivative of Planck function (radiance) with respect to brightness temperature.

34) VPLAN (T) T[K): Calculates Planck radiance from given temperature and channel.

35) VSKINT (TBB, JCOF, JUSE, JACT): Obtain surface skin temperature from VAS brightness temperatures. 'JCOF'=0 for empirical coefficients, 'JUSE'=0 to use day (2-channel) or night (3-channel) equation according to solar zenith angle, 'JUSE'=1 to use day equation, 2 to use night equation, 'JACT'=0 if nothing done, =1 if day equation actually used, =2 if night equation actually used.

36) WSAT (P, T): Calculates saturation mixing ratio for a given P and T.
Appendix V

Miscellaneous Program Information

This appendix will be of interest to users who wish to know about VAS retrieval programs in terms of core memory usage, code length, number of physical input-outputs (I-O's) during each execution, and (or) how many times a given program is called during the processing of a typical retrieval area.

The information for all the programs is arranged in a tabular format. The four columns headed 1-4 correspond to the above four pieces of information for each program. Column 1 contains core memory usage (in bytes * 1024, or "K" bytes). Column 2 holds program code length, including both total number of lines and number of comment statements. Column 3 lists the number of physical I-O's during a given program run, and column 4 indicates approximately how many times a given program would be executed during the processing of a typical retrieval area. It must be kept in mind that numbers 3 and 4 are only approximate figures, and can vary depending both on the particular data set being processed and the time at which the processing is being done.

Before the presentation of the table, a few comments on physical I-O's is in order. Physical I-O's are those I-O's which cause a "wait state" on the McIDAS system; that is, they cause system activity to be suspended until the input has been completed. For a given program, the following constitute physical I-O's:

1.) 1 I-O for program keyin
2.) 1 I-O for every page read off an LW (Large Word array) file
3.) 1 I-O for every track read off an area
4.) 1 I-O for every line which goes to a display device (TV or CRT line)

(Note: to see definitions of page, Large Word (LW) array file and track, turn to Appendix I, Section I.

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th><strong>1</strong></th>
<th><strong>2</strong></th>
<th><strong>3</strong></th>
<th><strong>4</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>VPVA</td>
<td>26</td>
<td>38=total</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32=code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDVA</td>
<td>164</td>
<td>348=total</td>
<td>54</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>290=code</td>
<td>in auto mode:47</td>
<td></td>
</tr>
<tr>
<td>LOVA</td>
<td>33</td>
<td>51=total</td>
<td>15</td>
<td>optional, usually .LE. 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>46=code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPVA</td>
<td>51</td>
<td>60=total</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>49=code</td>
<td></td>
<td>from 12 to 24, depending on # of keyword parameters used</td>
</tr>
<tr>
<td>GSVA</td>
<td>350</td>
<td>297=total</td>
<td>82</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>245=code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSVVA</td>
<td>61</td>
<td>228=total</td>
<td>230</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>187=code</td>
<td></td>
<td>depends on # of reports written into the sfc. MD file</td>
</tr>
<tr>
<td>SRVA</td>
<td>186</td>
<td>365=total</td>
<td>26</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>292=code</td>
<td></td>
<td>more if sfc. analyses must be redone</td>
</tr>
<tr>
<td>XRVA</td>
<td>117</td>
<td>269=total</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>230=code</td>
<td></td>
<td>depends on # of sfc. and (or) retrieval reports deleted</td>
</tr>
<tr>
<td>VTPZ</td>
<td>694</td>
<td>562=total</td>
<td>434</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>486=code</td>
<td></td>
<td>in auto mode:442 (when auto mode was done over a retrieval area approximately one-half the size of the &quot;442&quot; area above, I-O's only decreased to 411)</td>
</tr>
</tbody>
</table>
PLVA 121 324=total 73, when usually no more
277=code plotting approximately 200 sfc.
reports; 118, when plotting
approximately 195
retrieval 500 mb height
reports

BNVA 279 542=total 65 optional, depending upon
436=code how many grids
of retrieval
parameters are
desired (9, for
example, if
gradient winds
at standard
levels (850-100
mb) are
computed

SRAD 571 484=total 1998 l (not used
415=code in auto mode:2001 at present; VTPZ
(for retrieval area replaces SRAD)
of approximately one-
half the size of the
"1998" area: 1071)

SRET 257 799=total 1772 l (not used at
646=code present; VTPZ
(for same half-
area as above: 997) replaces SRET)

Note: physical I-O's for both SRAD and SRET depend heavily on
how many retrievals are to be processed in the retrieval area.

GWVA 159 288=total 325 9, for gradient
234=code winds at
standard levels
(850-300 mb)

UGVA 51 136=total 1806 1
98=code

EXVA 110 124=total 25, for 1 report,
113=code 46 for 5 reports;
that is, about 5
more I-O's for
each additional
report

GPVA 548 100=total 61 lesser-used
85=code program; used

360
APPENDIX VI

VAS Retrieval Software

The final appendix consists of copies of all the programs discussed in this manual, in the same order as listed in the Table of Contents. This appendix should be used in conjunction with a given Level I Flowchart and program description to achieve the best possible understanding of the program in question.
C FILE POINTER FOR VAS PROCESSING
C KEYIN: VPVA NN
C NN: SOUNDER AREA NO. (TC SET)
C NN=0 TO INTERROGATE
C :=-1 TO TURN OFF
C SSEC/'CRCAS USERS MANUAL - CHAP12
DIMENSION MIN(2),KOUT(10),IBLF(112),IDIR(64)
COMMON /ARENT,IDIR
CALL IO(MIN(2))
IF (''HELP'').EQ.LIT(''HELP'') OR MIN(2).EQ.LIT(''HELP'') GC TO 900
IF(FILE=MIN(2)
NF=FILE
IF(FILE.NE.90) GO TO 10
NF=9
GO TO 20
10 IF (FILE.GE.0) GC TO 200
IF (FILE.GT.5999) GC TO 800
20 CONTINUE
CALL PUS(NF,L1)
100 CALL TQMS(POINTER SET TO SOUNDER FILES*,NF)
IF(NF.LE.0) RETURN
CALL READD(NF,IDIR)
IF (IDIP(1).EQ.-1) GC TO 120
KOUT(1) = 3
KOUT(2) = IDIR(3)
KOUT(3) = IDIR(4)
KOUT(4) = IDIR(5)
CALL OUTINT(KOUT)
RETURN
120 CALL TQMS(POINTER NOT SET TO VALID SOUNDER FILES*,0)
RETURN
200 NF=LU(81)
GO TO 100
800 CALL TQMS(ILLEGAL POINTER REQUEST ...,*,0)
RETURN
E00 CALL TQMS(SNCR FILE NO.),-1 SETS POINTER TO OSS*,0)
RETURN
END
DIMENSION VAS(13), WFR(64), KCUT(10)
DIMENSION LAT(56), LCN(56)
DIMENSION IRET(300)
DIMENSION LEUF(33)
DATA LREF/33 240404040/
DATA LCOLS/56/
DATA LRET/300 780808080/
DATA LUN/20 /
DATA WLC/56C/ ELCK/C/ TCPP/360/ BOW/C/
DATA LRCF/0/ IEBC/5999999/
DATA WPSC/559999/
COMMON /NAV/FLAT, FLCN, ZENLOC, ZENIL, IE, IPAS, IFIC, IHMS, JT, JD
COMMON /CC/ICCC(112)
COMMON /AVT/ IDCIP(64)
COMMON ELCKS FCR VASDAT
COMMON/AUTC/IECX, IUP(6)
COMMON LAST/LASLIN, LASELE, LELE, ICHAR
MDR=IFF(1,0)
MDR=IFF(2,0)
LALGNW=IFF(3,0)
LALOSE=IFF(4,0)
KBUG=IKWP(*ELC*, 1,0)
IF (LUC(81) .EQ. 0) GO TO 10
FOLLOWING FCR VAS
ITERM=LLC(-20)
CALL DOPEN(*VASTEXT*, LUN, LEN)
CALL DREAD(LLA, ITERM, ICC)
GO TO 20
10 CALL TENIC(1, 1, 1, 1, 1, 1, ICC)
20 CONTINUE
IF (LALON.EQ.0) GO TO 110
IF (LALCSE.EQ.0) GO TO 930

C FOR TCVS FILL DIRECTORY AND EXIT
IDCC(25)=LALCSE
IDCC(26)=LALCSE
IF (LUC(81).EQ.0) GO TO 280
LAN=LALONW/1000
FLAT=LAN
LCW=MOD(LALCN,1000)
WLCN=LCW
IF (LCW.GT.180) LCW=LCW-360
FLON=-LCW

C USE NAV TO GET LINE,ELE
SNC=LUC(81)
CALL READ(NSD,ICIR)
JD=IDIR(4)+ICIR(3)/100000
JT=IDIR(5)
CALL NVIEW(EETAIN,EETDCT,INAV,FTIME)

C LOOK AT ALL CORNERS TO GET MAX AND MIN ELEMENT
CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLON+2,INAV,EETAIN,EETDOT).5
IL=FLIN+0.5
ITE=FELE+0.5
LAS=LALCSE/1000
FLAT=LAS
CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLON+2,INAV,EETAIN,EETDOT).5
IEE=FELE+0.5
IEE=MIN(ITE,IEE)
LOE=MOD(LALCSE,1000)
ELON=LOE
IF (LOE.GT.180) LOE=LOE-360
FLON=-LOE
CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLON+2,INAV,EETAIN,EETDOT).5
LL=FLIN+0.5
IEL=FELE+0.5
FLAT=LAN
CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLON+2,INAV,EETAIN,EETDOT).5
IEL=FELE+0.5
IEL=MAX(ITE,IEE)
ILF=IDIR(12)
IER=IDIR(13)
GO TO 115

100 IF (LUC(81).EQ.0) GO TO 280

C TRANSFER FOR TCVS
MFR=LUC(-1)

C PICK UP IMAGE INFO
CALL GETFRM(MFR,MFR)

C LOAD TIME PARAMETERS IN COMMON /NAV/
JC=MFR(1)*100000+MFR(2)
JT=MFR(2)
IL=MFR(5)
IE=MFR(6)

C IL AND IE GIVE AREA LINE AND ELEMENT OF UPPER LEFT DATA
105 C DISPLAYED ON TV FRAME (NOT NECESSARILY RAS/FIX 1.1)
106 IX=IL
107 IX=IE
108 MAG=MFR(10)
109 IRL=MFR(11)
110 IER=MFR(12)
111 C LAST ARE LINE AND ELEMENT RESOLUTION
112 C PICK UP SPLINTER AREA RC
113 NSINC=MFR(17)
114 IDOC(3)=IL
115 IDOC(4)=IE
116 C CALCULATE AREA COORDINATES OF LOWER RIGHT RAS/FIX ON TV SCREEN
117 LR=LUC(11)
118 LF=LUC(12)
119 LL=I Perry
120 LE=IX++(LP-MFR(9)*IER)/MAG
121 IF (KBLG.AEO.0) CALL EKCCDE("IL+IE+ILR+IER"+IIE")+LEUF,
122 IL+IE+ILR+IER)
123 C ENSURE INITIAL ELEMENT IS OK
124 C AND GET LAT AND LONG OF POSITION
125 IEB=(IE+LE)/2
126 IF (KBLG.AEO.0) CALL EKCCDE("LL+LE+IEB",3IP")+LBFL,
127 LL+LE+IEB)
128 C USE VASDAT TO PICK UP BOUNDS OF DATA ON TV SCREEN OR AS FORCED
129 C BY LAT AND LONG ENTRIES
130 C PASS FLAGS TO VASDAT THRU COMMON
131 IBOX=0
132 IASEL=-1
133 IASLINE=-1
134 ISEC=0
135 C USE ISEC TO INSIST ON 4 CONSECUTIVE LINES OF DATA. THIS
136 C IS TO POSITION SPACING OF RETRIEVALS IN SMALL DETECTOR MODE
137 120 VAS(1)=0.
138 C VAS(1) MUST BE NON NEG TO RETURN DATA VECTOR
139 C CHECK TCP LINE
140 C CALL VASDAT(IL+IEE.VAS)
141 LAN=FLAT+0.5
142 IF (VAS(1).GT.0.AND.VAS(7).NE.V*ISG) GO TO 125
143 VAS(7) IS SMALL DETECTOR...MAYBE
144 ISEC=0
145 122 IL=IL+ILR
146 IF (IL+GT.0) GC TO 300
147 GC TO 120
148 125 ISEC=ISEC+1
149 IF (ISEC.NE.1) GO TO 126
150 ILS=IL
151 LAN=FLAT+0.5
152 126 IF (ISEC.NE.4) GC TO 130
153 GC TO 122
154 130 VAS(1)=0.
155 IL=ILS
156 IETCP=IE
LASLIN = 1
VAS(1) = 0.

CALL VASCAT(IL, IETCF, VAS)
IF (FLCN. LT. 0.)FLCN = FLCK + 360.
IF (VAS(1). GE. 0. .AND. FLCK. LE. 0.)GO TO 159
C LONG TEST IS TO CORRECT PARALLAX
IETCF = IETCF + IER
IF (IETCF. GT. LE) GO TO 300
GO TO 133

TCP = FLON
C CHECK LAT/LCN FOR CE
IF (LALCN. LE. 0.) GO TO 138
C MOVE ACROSS TC TO GET EASTERN/CST ELEMENT
LETCF = IETCF

LETCF = IETCF + IER
VAS(1) = 0.
CALL VASCAT(IL, IETCF, VAS)
IF (FLCN. LT. 0.)FLCN = FLON + 360.
IF (VAS(1). GT. 0. .AND. FLCK. GT. ELON) GO TO 136
TCPE = FLON
C PICK UP INITIAL TIME FROM SOUNDER FILE

ICOC(10) = IHMS
CALL SDEST(\ldots; LPF, LEFT DCNE*, 0)
C GC TO LOWER RIGHT CORNER (OF TV) AND BACK UP TO DATA
LASLIN = -1
CONTINUE

VAS(1) = 0.
CALL VASCAT (LL, IEE, VAS)
IF (VAS(1). GE. 0.) GO TO 150
LL = LL - ILR
IF (LL. LE. IL) GO TO 400
GO TO 140
CONTINUE

LAS = FLAT - 0.5
LOE = FLCN - 0.5
LERC = LE
IF (LALCN. LE. 0.) GO TO 156
C MARCH ALONG LINE TO AGAIN GET ELEMENT NCS, WITH LAT/LON BOUNCES
IEBCT = IE

LASLIN = 1
VAS(1) = 0.
CALL VASCAT(LL, IEBCT, VAS)
IF (FLCN. LT. 0.)FLCN = FLCK + 360.
IF (VAS(1). GE. 0. .AND. FLCK. LE. 0.)GO TO 159
IEBOT = IEBCT + IER
IF (IEBCT. GT. LE) GO TO 400
GO TO 153

EGCT = FLON

VAS(1) = 0.
CALL VASCAT(LL, LEBCT, VAS)
IF (FLCN. LT. 0.)FLON = FLON + 360.
IF (VAS(1). GE. 0. .AND. FLON. LE. GE. ELON) GO TO 157
LEBCT=LEBCT+IF
210  IF (LEBCT.LT.1E) GO TO 400
211  CC TO 156.
212  158 BOTE=FLOC
213  IE=INT1(1ETCF+IEBCT)
214  LE=MAX(LETCF,LEFCT)
215  LCW=AMAX1(TCFL,ECTW)+0.1
216  LOE=AMAX1(TCPE,ECTE)-0.1
217  158 IF (LCT.LT.C) LCE=360.+LCE
218  IF (LCD.LT.C) LCW=360.+LCW
219  IF (KREGS.GE.C) CALL ENKDE(1X,"TCPL,TCPE,ECTL,ECTE",4F7.2)
220  LEUF,TPW,TCPE,ECTW,BOTE)
221  CALL SEDST(*.LCW,RRG,RIGHT,DOE.,0)
222  IDCC(11)=IFMS
223  DO 160 L=1,13
224  IDCC(11+L)=C
225  IF (VAS(L).AM.GE.99999.) IECC(11+L)=1
226  160 CONTINUE
227  LALCNW=IAES(LAN)+1000+LCW
228  IF (LAN.LT.0) LALCNW=-LALCNW
229  LALCSE=IAES(LAS)+1000+LCE
230  IDCC(25)=LALCNW
231  IF (LAS.LT.0) LALCSE=-LALCSE
232  IDCC(26)=LALCSE
233  CALL SEDST(*.LINUL,LINUL,LATUL,LCNUL,LINLR,ELULR)
234  TLR, LONL, 0)
235  KOUT(1)=R
236  KOUT(2)=IL
237  KOUT(3)=IE
238  KOUT(4)=LAN
239  KOUT(5)=LCW
240  KOUT(6)=LL
241  KOUT(7)=LE
242  KOUT(8)=LAS
243  KOUT(9)=LCE
244  CALL OUTINT(KOUT)
245  C FILL VASRET CCCUMENTATION FROM SCURDIR DIRECTORY
246  IDCC(1)=ICIR(3)+100000+ICIR(4)
247  DC 170 I=2+11
248  170 IDCC(1)=ICIR(1+3)
249  IF (MDNR.EG.C) GC TO 260
250  IF ("CPCN(MCNR.2).LT.0") GC TO 900
251  C INITIALIZE RETRIEVAL MC RCE
252  IF (MDNR.EG.C) GC TO 250
253  IRET(1)=MCC(ICCC(1),100000)
254  IHF=IDCC(2)/10000
255  IRET(2)=ICCC(2)
256  IRET(3)=0
257  ICK=MCPUT(MDNR,MDCR.0,CRET)
258  IF (ICK.LT.0) GC TO 910
259  250 CONTINUE
260  IDCC(40)=MCNR
IDCC(43)="*ERR"
CONTINUE
C SET STAILS FLAG TC READY
IDCC(26)=1
C PUT IN SNSR FILE NC
NSNC=LUC(65)
IDCC(35)=NSNC
C SET LINE AND ELEMENTS FOR SRAD
IDCC(50)=0
IDCC(51)=0
C SET SPACING DEFAULTS FOR SRAD
IDCC(52)=IC*16/ILP+0.5
C FOLLOWING IS FOR ECX SIZE (FER RETRIEVAL...E5 FOR PES 16)
IDCC(53)=5*16/ILP+0.5
C LOOK FOR FORCED OVERRIDE
IDCC(53)=IMLF("IBCX",1,IDCC(53))
C INDENT BY HALF THE BCX SIZE
ICFF=IDCC(53)/2
IF (MCC(ICFF+2).*NE.*1) ICFF=ICFF+1
ICFF=ICFF ILR
IDCC(54)=0
IDCC(55)=LL-ICFF
IDCC(56)=LE-ICFF
IDCC(57)=1L+ICFF
IDCC(58)=IE+ICFF
IDCC(59)=ITERM
IDCC(60)=0
CALL DWRITE(LLN,ITERM,IDCC)
CALL DCLOSE(LUN)
CALL SDEST("* DOCUMENTATION COMPLETE FOR TERMINAL *,ITERM,
RETURN
CONTINUE
C BEGIN SEARCH FOR MAX AND MIN LAT AND LNG REQURED BY ORBIT
C WATCH FOR MISSING LINES AT BEGINNING AND END OF ORBIT
MCPW=IDCC(4)
DC 2 N=5+NRCW
K=N
CALL TSNIC(C+1,K+1,1,NCCLS,LAT)
IF (LAT(1).LE.5000.AND.LAT(NCCLS).LE.5000) CC TC 3
CONTINUE
CALL TSNIC(C+2,K+1,1,NCCLS,LON)
MODE=1
IF (LAT(NCCLS).LT.LAT(1)) MODE=2
C MODE=2 IS DESCENDING
MAXLAT=LAT(NCCLS)+300
MAXLON=LON(1)-500
IF (MODE.*EG.*1) CC TC 4
MAXLAT=LAT(1)+300
MINLON=LON(NCCLS)+500
NR=NROW-5
DC 5 N=NR*1,-1
CALL TSNIC(C, 1, K, 1, 1, NCCLS, LAT)

IF (LAT(1) LE. 5000 OR LAT(NCCLS) LE. 9000) GO TO 44

5 CONTINUE

44 CALL TSNIC(C, 2, K, 1, 1, NCCLS, LCA)

IF (NODE = 2) GO TO 6

MINLAT = LAT(1) - 300

MINLON = LCA(NCCLS) + 500

GO TO 7

2 MAXLON = LCA(1) + 300

MINLAT = LAT(NCCLS) - 300

7 CHANGE LCLGHT TO LONG CONV PCs & NEG E

MAXLON = MAXLON

MINLON = MINLON

CHANGE TO 0 - 360 W FOR STORAGE

IF (MAXLON LT 0) MAXLON = MAXLON + 36000

IF (MINLON LT 0) MINLON = MINLON + 36000

MAXLAT = MAXLAT/100

MINLAT = MINLAT/100

LALCNW = IABS(MAXLAT) - 1000 - MAXLON/100

IF (MAXLAT LT 0) LALCNW = -LALCNW

LA = IABS(MINLAT) - 1000 - MINLON/100

IF (MINLAT LT 0) LALCSE = -LALCSE

IDCC(25) = LALCNW

IDCC(26) = LALCSE

IDCC(40) = MENR

IDCC(41) = MRWR

C SET STATUS FLAG TO READY

IDCC(26) = 1

CALL TSNIC(3, 1, 1, 1, 1, 1, 1, IDCC)

4 CALL SDEST('...FRCPFA LCCATING NW CORNER...

300 CALL SDEST('...FRCPFA LCCATING NW CORNER...

KEYIN*0)

4 CALL ABOERT

4 CALL SDEST('...FRCPFA LCCATING SE CORNER...

500 CALL SDEST('...CANNOT OPEN RETRIEVAL MD FILE NC', MENR)

510 CALL SDEST('...CANNOT WRITE MD ROW HEADER FOR RCW', MRWR)

520 CALL SDEST('...LALCSE HAS NOT BEEN SPECIFIED*', 0)

5 CALL SDEST('...FRCPFA LCCATING NW CORNER...

END
C ? TO QUERY VASTEXT DOCUMENTATION SECTOR (IF VAS POINTER SET)
C ? CR TOCCRP TO WHICH TOCS POINTER IS SET
C ? KEYIN: LCOVA -NO PARAMETERS-
C SSEC/UCDAS USERS MANUAL - CHAP12
DIMENSION VAS(13),IEBF(112),MFR(64),KOUT(10)
DIMENSION TGELF(20)
COMMON /CCOC/IDCC(112)
CHARACTER EQ TITLE,TIT2,TIT3,TIT4
REAL *8 FILE
DATA TITLE/* *VCDD BEGIN Y-RES X-RES LLAW LLSE STAT
DATA TIT2 /* MCAS MCSR MDNG MDRG MCNR MDRR
DATA TIT3 /* NGFS ZCRC TGRID CGRID
DATA TIT4/* TYP GSS SFC SIZ SFC ENCL ENDE BEGL BEND
DATA LUN/2C/* FILE/VASTEXT /* LEN/100/
COMMON /NAV/FLX,FLON,SENLOC,SENIL,IE,IRAS,IFIC,IFMS,JT,JD
COMMON /APET/TEIR(64)
IF (LUC(81).EG.0) GC TO 1
C FOLLOWING FOR VAS
ITER=LUC(-20)
CALL DCPEM(FILE,LUN,LEN)
CALL DREAD(LLA,ITER,ICOC)
NRET=IDCC(1C)
NSAT=IDCC(1)/10CCG
GO TO 2
CALL TSNIC(1,1,1,1,1,1,1,ICOC)
CALL RETIO(IELF,0)
APET=IBLF(112)
NSAT=IDCC(3)
IDCC(5)=IDCC(4)
IDCC(10)=56
2 CALL SCESD(TITLE,0)
IYC=MCD*IDCC(1)/100000
ITIM=IDCC(2)
NSND=IDCC(25)
CALL ENKODE(* (416,217,16,17,19/*),TGELF,IYC,ITIM,
1 IDCC(5),IDCC(10),IDCC(2E),IDCC(26),IDCC(28),NSAT,NSND)
CALL SCESD(TIT2,0)
CALL ENKODE(* (416/*),TGELF,IDCC(2E))
CALL SCESD(TIT3,0)
CALL ENKODE(* (416,418/*),TGELF,IDCC(42),IDCC(29))
CALL SELST(*,NC.RETRIEVALS=*,MRCF*)
IF(LLC(FL),EG,6)RETURN
CALL SELST(*,CURRENT RETRIEVAL CFTICKS=*,0)
CALL SELST(TIT4,C)
CALL ENKODE(*,(1116/5),TCDUF,IDCC(50))
RETURN
END
SSECC/"CICAS USERS MANUAL - CHAP12
COMMON /CCCI/IDCC(12)
COMMON /TERMAL/ITERM
DIMENSION IRET(300),CPAR(10)
CHARACTER*12 CPAR
DATA LUN/207,LEN/100/
DATA CPAR/*MCN*,*MERS*,*MCNG*,*MCRG*,*MDNR*,*MCR*,
*NGFS*,*ZGRIC*,*TERIC*,*DGRIC*/
KBUC=IKWF(*ELC*,1,0)
IF (LUC(81),EG,0)GO TO 1
C FOLLOWING FCR VAS APPLICATIONS
ITERM=LUC(-20)
CALL DOPEN(*VASTEXT,*LUN,LEN)
CALL DREAD(LUN,ITERM,IDCC)
GO TO 2
CALL TSNIC(1,1,1,1,1,1,1,IDCC)
2 CONTINUE
C CHECK LAT LCA CCVEPAGE
LLNW=IKWF(*LLNW*,1,0)
LLSE=IKWF(*LLSE*,1,0)
IF (LLNW,NE,0)IDCC(25)=LLNW
IF (LLSE,NE,0)IDCC(26)=LLSE
C CHECK MD NC*
DO 10 M=1,6
MESS=IKWF(CPAR(M),1,0)
IF (MESS,EG,0)GO TO 10
IDCC(35+M)=MESS
10 CONTINUE
C CHECK GRIC CCC
DO 20 M=7,10
MESS=IKWF(CPAR(M),1,0)
IF (MESS,EG,0)GO TO 20
IDCC(22+M)=MESS
20 CONTINUE
MESS=IKWF(*NGFS*,1,0)
IF (MESS,NE,0)IDCC(42)=MESS
IF (LUC(81),EG,0)GO TO 25
CALL DWRITE(LLN,ITERM,IDCC)
CALL DCLOSE(LLN)

54  GO TO 28

55  CALL TSNNIC(3,1,1,1,1,1,IDCC)

56  C  CHECK CN RETRIEVAL AN. EDIT

57  28  NRET=IKWP(*NRET*,1,-1)

58  IF (NRET=EG-1) GC TO 30

59  IF (LUC(81).EQ.0) GC TO 35

60  CALL VRTIC(IRET*,0,0)

61  IRET(3)=NRET

62  IDOC(160)=NRET

63  CALL VRTIC(IRET*,0,1)

64  GC TO 30

65  35  IDOC(112)=NRET

66  CALL RETIC(IDCC,0,1)

67  30  IF(KBLG.GE.0)CALL SDEST(* DUN....,0)

68  RETURN

69  END
C PROGRAM TO PREPARE MD FILE OF GUESS PROFILES (CMF)
C KEYIN: GSV A NGR MDOM LLNW LLSE INC NGRP CELT
C
C POSITONAL PARAMETERS:
C NFG - GRID FILE OF GUESS GRIDS
C NGF - FIRST GRID OF LATER (LATEST) SET OF GUESS GRIDS
C MNC - MCDF OUTPUT ARRAY
C LLW - LAT LLF FOR MD FILE (0-360W)
C LLN - SE LAT LCF OR MD FILE
C INC - LAT/LLC INCREMENT IN DEGREES
C NGRF - FIRST GRID OF EARLIER SET FOR TIME INTERP
C CELT - FRACTIONAL TIME INTERP (EARLY+CELT/(LATE-EARLY))
C
C USER/MICMAS USER MANUAL - CHAP12
COMMON /DOC/IDOC((112)
DIMENSION LIST(26),KCUT(10),ISCALE(26),UNIT(26),LOCs(26)
C ABOVRE TO RECEIVE MKY INFO
DIMENSION IGRC(3200,23),MES(64),JREC(26),IREC(26),FISC(22)
DIMENSION ICAR(22),KCAR(22),JCAR(22),IFSSS(22),IGOT(22)
DIMENSION LOCAR(22)
INTEGER IGCF(64),IGOF(64),IGID(8),RCW(3),COL
C------DESCRIPTION OF IGCF=MD-FILE GRID HEADERS
C
C GIVE TOTAL SIZE (WCDF), # ROWS, # COLS. (IGSF=IGNR,IGNC)
C EQUIVALENCES (IGF=ICF(1)),(NIGF,IGHC(2)),(IGNC,IGHC(3))
C YMD.OY, YYMM.SS AND VALIC-TIME (IF APPLICABLE) FOR GRID
C EQUIVALENCES (IGCY,IGHC(4)),(IGTMV,IGHC(5))
C DESCRIPTION OF GRID VARIABLE (IN MD-FILE TERMS):
C NAME, SCALE, AND UNITS
C EQUIVALENCES (IGK,V,IGHC(7)),(NIGVSCA,IGHC(8)),(IGNVNS,IGHC(9))
C DESCRIPTION OF VERTICAL LEVEL: VALUE, SCALE, AND UNITS
C EQUIVALENCES (IGLEV,IGHC(10)),(IGSCL-A,IGHC(11)),(IGLNA,IGHC(12))
C-GRID-DECE-VALUE TYPE == 1 (TIME DIF) 2 (TIME AVG) 4 (LEVEL DIF)
C 8 (LEVEL AVG) OR ANY SUM OF THE FOREGOING
C EQUIVALENCES (IGTYF,IGHC(13))
C FOLLOWING USE IF PARAMETER IS A VERTICAL (LEVEL) DIF OR AVG
C SAME SCALE AS IGLLEV
C EQUIVALENCES (IGLDIF,IGHC(14))
C FOLLOWING USE IF PARAMETER IS A TIME DIF OR AVG (YYMMSS)
C EQUIVALENCES (IGTDIF,IGHC(15))
C GRID ORIGIN, TYPE (E.g. TYPE OF PROJECTION)
C EQUIVALENCES (IGCRG,IGHC(33)),(IGCTYPE,IGHC(34))
C SUBSEQUENT CCPRDS (IGLCM,IGLCM), ALL HAVE 4 IMPLIE DEG. PLACES.
C LAT GOES FROM 500000 TO 900000, LON GOES FROM -1800000
C----- TO 1800000 (WEST IS *)
C----- TYPE 1 GRIDS ARE PSEUDOC-MERCATOR
EQUIVALENCE (IGLAM*,IGEC(35)),(IGLCR*,IGEC(36)),(IGLAM*,IGEC(7))
     *(IGLCR*,IGEC(38)),(IGINC*,IGEC(35))
C----- TYPE 2 GRIDS ARE POLAR-Stereographic
C----- GIVE RCW # CF NORTH POLE, COL # CF N.P., COL SPACING AT 60 DEG.
C----- (DEG = LATITUDE PARALLEL TO COLUMNS (DEG)
EQUIVALENCE (IGPCLP,IGEC(35)),(IGPCLC,IGEC(36)),(ICSF65,IGEC(37))
     *(IGLCR*,IGEC(38))
C----- INITIALS OF USER AND PROJECT # UNDER WHICH GRID CREATED
EQUIVALENCE (ICLSSB,IGEC(41)),(IGPCRC,IGEC(42))
C----- CHARACTER ID SUPPLIED BY PROGRAM (ARBITRARY)
EQUIVALENCE (IGICT,IGEC(43))
EQUIVALENCE (ICOL,IREC(1))
EQUIVALENCE (COL,IREC(4))
DATA IFRECC/1000.250.700.500.400.300.250.200.150.100.70.50.30.20.10.
     10.1000.850.700.500.400.300.1000/
70 C FOLLOWING 2 SPECIAL CASES FOR CALLAN'S ANMRC AND ECMWF FIASCCS
71 DATA LCCHAR/15. "TYP".6."DPT","HT"
72 DATA ICCHAR/15.*TEMP.*5.*TDPT.*MHT*/
73 DATA KCCHAR/15.*T.*6.*T.*2.*Z*/
74 DATA JCCHAR/15.*T.*6.*TD.*2.*Z*/
75 DATA ICCT/22.0/
76 DATA MISG/28080080/
77 DATA LUN/2/.*LEN/100/
78 C
79 C
80 C
81 NGF=FF(1.0)
82 IF (LUCC01. EG=0) GO TO 110
83 ITE=LUCC(20)
84 CALL DOPEN(*VASTEXT*,LUN,LEN)
85 CALL DREAD(LMN,ITERM,IDCC)
86 GO TO 120
87 110 CALL TSNIC(1,1,1,1,1,1,1,IDOC)
88 120 CONTINUE
89 DC 1 N=1.*22
90 DC 1 I=1.*3200
91 1 IGRID(I+N)=MISG
92 NGB=FF(2.1)
93 MDNG=FF(3.0)
94 IF (MDCPEN(MCNG.2).LT.0) GO TO 960
95 LLNW=IFF(4.0)
96 IF (LLNW. EG=0) LLNW=IDCC(25)
97 LLSF=IFF(5.0)
98 IF (LLSF. EG=0) LLSF=IDOC(26)
99 INC=IFF(6.0)
100 IF (INC. EG=0) INC=20
101 KEUG=IKWF(*ELG*.1.0)
102 1 READ IN ENTIRE SET OF GUESS GRIDS
103 C GET KEYS TO ESTABLISH SCALING
104 ICK=MCKEYS(MCNG-.1.*LIST.*ISCALE.*ILNIT.*LCCS)
105 IF (ICK·LT·0) GO TO 920
106 C IMPLICIT LCOF OVER TWO TIME PERIODS
107 100 NGE=NG5+70
108 L=0
110 DO 10 K=AGE·AGE
111 ICK=IGGET(NCPG*K·3200,IGRID(1,23),NR,NC,IGHCT)
112 C FOLLOWING NECESSARY BECAUSE OF GRIDS WITH HOLES
113 ITVNAME=IGHCT(7)
114 ITLEVEL=IGHCT(10)
115 IF (ICK·EG·0) GO TO 2
116 IF (ICK·LT·-1) GO TO 930
117 GC TO 10
118 CONTINUE
119 C ORDER THE GRIDS FOR MCFILE
120 NFR=NR NC
121 DC 3 I=1,22
122 JE=I
123 IF (LCHAR(I)*EG·ITVNAME) GO TO 4
124 IF (ICHAR(I)*EG·ITVNAME) GO TO 4
125 IF (KCHAR(I)*EG·ITVNAME) GO TO 4
126 IF (JCHAR(I)*EG·ITVNAME) GO TO 4
127 3 CONTINUE
128 GC TO 10
129 4 DO 5 J=JE·22
130 IF (IGCT(J)*EG·1) GO TO 5
131 L=J
132 IF (IPRESS(L)*EG·ITLEVEL) GO TO 6
133 IF (MOD(J,1E)*EG·1 AND·ITLEVEL·EG·500) GO TO 6
134 5 CONTINUE
135 GC TO 10
136 6 DO 8 J=1,NFT
137 IGRID(J,L)=IGRID(J,23)
138 8 CONTINUE
139 DO 80 I=1,64
140 80 ICHD(I)=IGHCT(I)
141 IRSAV=KR
142 MCSAV=NC
143 KCUT(1)=7
144 KCUT(2)=IGDAY
145 KCUT(3)=ITIME
146 KCUT(4)=ITLEVEL
147 KCUT(5)=L
148 KCUT(6)=IRSAV
149 KCUT(7)=MCISA
150 KCUT(9)=IGRID(100,23)
151 IF (KBUG·IE·C) CALL CUTINT(KOUT)
152 C SET UP FACTOR BETWEEN GRID SCALING AND MD SCALING
153 ICON=ISCALE(L+4)-IGVSCA
154 SCL=10.-·ICNA
155 FISC(L)=SCL
156 IGOVT(L)=1

377
157  NCCT=0
158  CC 11 J=1,22
159  10 NCCT=NCCT+ICCT(C)
160  IF (NCCT.EQ.22) GO TO 12
161  10 CONTINUE
162  IF (L*EG.*0) GO TO 550
163  12 CONTINUE
164  CALL SDEST(‘ PREPARING GLESS FOR DAY ’*IGDAY)
165  CALL SDEST(‘ HOUR = ’*IGTIME)
166  CALL SDEST(‘ VALID TIME = ’*IGTIME)
167  GET DATE INFORMATION FROM GRID AND MOVE TO ROW HEADER
168  ROW(1)=IGDAY
169  ROW(2)=IGTIME
170  FGINC=FLCAT(IGINC)
171  LAN=LLNW/1000
172  LAS=LLSE/1000
173  NLA=(LAN-LAS):10/INC+1
174  LNW=IABS(LLNW/1000)
175  LLCE=IABS(LLCE/1000)
176  IF (LCE.GT.LCW) LCE=LCE-360
177  IF (LCW.GT.LCE) LCW=LCW-360
178  IF (LCE.GT.LCW) LCE=LCE-360
179  NLC=(LCW-LCE)*10/INC+1
180  NLAT=LAN 1000
181  JINC=INC 1000
182  IF (KBLG.EQ.0) GO TO 45
183  CALL SDEST(‘ IGLAMX IGLAMN IGLCMX IGLC MN IGINC ’,0)
184  KCUT(1)=5
185  KCUT(2)=IGLAMX
186  KCUT(3)=IGLAMN
187  KCUT(4)=IGLC MX
188  KCUT(5)=IGLC MN
189  KCUT(6)=IGINC
190  CALL OUTFIL(KCUT)
191  KCUT(2)=LAN
192  KCUT(3)=LAS
193  KCUT(4)=LCW
194  KCUT(5)=LCE
195  KCUT(1)=4
196  CALL OUTFIL(KCUT)
197  CALL SDEST(‘ NLA = ’*NLA)
198  CALL SDEST(‘ NLC = ’*NLC)
199  CONTINUE
200  IF (IGLC MX.LT.0.AND.IGLCMN.GT.0) IGLCMX=36000000+IGLC MX
201  MN=0
202  CC 50 V=1,NLA
203  IF (NLAT.GT.IGLAMX) GO TO 50
204  IF (NLAT.LT.IGLAMN) GO TO 50
205  FU=FLOAT(IGLAM X-NLAT)
206  SU=FU/FGINC+1.0
207  J=SU
208  DJ=SU-J
ORDER

JF=J+1
IF (JF.GT.APSAV) JF=J
RCE(3) = NLAT
WRITE RCE HEADER
NN=NN+1
ICK=MDFUT(MCNC,NN,C,IPEC)
IF (ICK.LT.C) GC TC 570
LON=LOW 10000
M*=0
DO 40 M*=1,KLC
IF (LON.GT.IGLCMX) GC TC 40
IF (LON.LT.IGLCMN) GC TC 40
FI=FLCAT(IGLCMX-LCN)
SI=FI/FGIN6+1.0
I=SI
DI=SI-I
IF (I=I+1
IF (IF.GT.ACSAV) IF=I
L=(I-1)*NRSAV+J
L2=(I-1)*NRSAV+J
L3=(I-1)*NRSAV+J
L4=(I-1)*NRSAV+J
M*=MM+1
IF (NN.NE.1) GC TO 7
COL=LON
IF (CCL.GT.1800000) CCL=CCL-3600000
WRITE COL HEADER
ICK=MDFUT(MCNC,0,MM,IPEC)
IF (ICK.LT.C) GC TO 580
WRITE DATA
7 DO 9 K=1,22
IVAL=IGRID(L,K)
IF (IVAL.EQ.MISC) GC TO 9
TL=IGRID(L,K)
TR=IGRID(L2,K)
BL=IGRID(L3,K)
EP=IGRID(L4,K)
TVAL=TL+CI*(TR-TL)
EVAL=BL+CI*(EL-EL)
VAL=TVAL+CI*(EVAL-TVAL)
VAL=VAL FISC(K)
IVAL=IVAL
9 IREC(4+K)=IVAL
C CHECK ON MISSING 200E T (HARRIS)
IF (IREC(16).EQ.MISC) IREC(16)=(IREC(17)+IREC(15))/2
DO 35 L=1,21
K=L+4
IF (IREC(K).EQ.MISC) GC TO 35
IF (IREC(K).LT.15000) IREC(K)=IREC(K)+27312
CONTINUE
35 IF (NBEF.EQ.C) GC TO 30
ICK=MDFUT(MCNC,NN,MM,JREC)
DO 28 L=1,22
262 K=L+4
263 FINC=GT FLCAT(JREC(K)-IREC(K))
264 IREC(K)=IREC(K)+FINC
265 CONTINUE
266 28 CONTINUE
267 C CHECK FOR CENTIGRADE (HARRIS GRIDS)
268 ICK=MIPUT(MCKC,AK,PK,IREC)
269 IF (ICK.LT.0) GO TO 990
270 40 LC(N)=LC(N)-JINC
271 50 MLAT=MLAT-JINC
272 IF (MGF.NE.0) GO TO 65
273 MGF=IPF(7,0)
274 NGENF=1
275 DT=CPP(8,0)
276 DC 60 K=1,22
277 60 JGOT(K)=0
278 IF (MGF.NE.0) GO TO 100
279 65 CALL MCLCLS(MFNG)
280 IDCC(33)=ICDAY
281 IDCC(34)=IGTIME 100+IGTIMK
282 IDCC(35)=MCNG
283 IDCC(36)=MEPG
284 IDCC(42)=MGFG
285 IF (LC(N).NE.0) GO TO 70
286 CALL CWRITE(LLN,ITERM,IDCC)
287 CALL FCLCSE(LLN)
288 GO TO 75
289 70 CALL TSNIC(3,1,1,1,1,1,IDCC)
290 75 CALL SDEST("RUN ",0)
291 RETURN
292 920 CALL SDEST("UNABLE TO GETEAM MC KEYS ",0)
293 RETURN
294 930 CALL SDEST("UNABLE TO OPEN Grid FILE NC. ",NGFC)
295 RETURN
296 950 CALL SDEST("CANNOT FIND ONE SINGLE LOUSY GRID... NOT CNE",0)
297 RETURN
298 960 CALL SDEST("UNABLE TO OPEN MC FILE NO. ",MDNG)
299 RETURN
300 970 CALL SDEST("CANNOT WRITE RCW HEADER ",R)
301 RETURN
302 980 CALL SDEST("CANNOT WRITE COL HEADER ",CM)
303 RETURN
304 990 CALL SDEST("CANNOT WRITE CARA VECTOR ",L)
305 RETURN
306 END
53   ITM=IDOC(2)
54   IF(INR.EC.x.1)IH=ITM/10000
55   IC=IC-1+ICCC(1)
56   IC=MOD(ICC(1)+1000)
57   IH=IC+1
58   CONTINUE
59   FL=IKW(9,3,F.1.
60   IF(NFL>0)GO TO 11
61   C     ASSUME FILE IS IN CURRENT SECTION
62   FL=MOD(3,10)
63   IF(NF<10,NFL=NF+10)
64   IF(KBUC.x.0,0)NFL=NFIL+30
65   IF(KBUC.x.0,0)NFL=NFL+30
66   IF(MDAT(IFC)(NFL,NE.0,0))GO TO 970
67   IC=MOD(1,NFIL)
68   IF(IOPN LT.0,0)GO TO 970
69   C     PICK UP KEYS
70   IC=MODKEYS(IFIL.13,16,ISL,IMITS,LOC)
71   IF(KBUC.x.0,0)GO TO 12
72   DO 85 J=1,9
73  85   KOUT(J+1)=LCCS(J)
74   KOUT(1)=8
75   CALL OUTINT(KOUT)
76   DC 66 J=1,9
77  86   KOUT(J+1)=ISCL(J)
78   KOUT(1)=8
79   CALL OUTINT(KOUT)
80   DO 83 J=1,9
81  83   KOUT(J+1)=LCCS(J+2)
82   KOUT(1)=4
83   CALL OUTINT(KOUT)
84   DO 88 J=1,9
85  88   KOUT(J+1)=ISCL(J+2)
86   KOUT(1)=4
87   CALL OUTINT(KOUT)
88   CONTINUE
89   C     SET UP DEFAULT LAT/LCN LIMITS
90   LLNN=ITOC(2)
91   LLNE=ITOC(2)
92   LISC=LLNN+1000
93   LAN=IKW(9,2,LAN)
94   MAXLAT=LNN+100
95   LAS=LISC+1000
96   LASC=IKW(9,1,LAS)
97   MINLAT=LNN+100
98   LCN=MOD(LLL,1000)
99   LCN=IAFW(LCN)
100  IF(LCN=30)LCN=LNN+360
101  LCN=IKW(9,2,LCN)
102  MAXLCN=LCN+100
103  LCN=MOD(LCN,1000)
104  LCN=IAFW(LCN)

382
IF (LCE.GT.180) LCE = LCE - 360
LCE = IKUP (*LCE*, 1, LCE)
MINLOK = LCE 1CC
CALL SCESS(* MAXLT MINLT MAXLA MINLA *, 0)
KOUT(1) = 4
KOUT(2) = MAXLAT
KOUT(3) = MINLAT
KOUT(4) = MAXLCA
KOUT(5) = MINLCA
CALL OUTINT(KOUT)
SET UP INDICES
JTYF = LCCS(1)
JDAY = LCCS(2)
JTIM = LCCS(3)
JLAT = LCCS(4)
JLON = LCCS(5)
JHMS = LCCS(6)
JELV = LCCS(7)
JTE = LCCS(8)
JDEW = LCCS(10)
JDIR = LCCS(11)
JSPC = LCCS(12)
JPRE = LCCS(13)
DO 32 I = 1, 72
SEARCH ROW HEADER FOR TIME AND DAY... UNLESS WE ARE FORCING FIRST AVAILABLE ROW AS IN AUTO PROCESSING
IOK = MDGET(AFIL, I, 0, IBUF)
IF (IOK .NE. 0) GO TO 32
ISV = 1
ITIME = IBLF(JTIM) / 10000
IF (MOD(IBLF(JDAY), 1000) .NE. IDAY) GO TO 32
IF (IKW .NE. 0) GO TO 35
31 IF (IHR .EQ. ITIME) GC TO 35
32 CONTINUE
GO TO 900
35 KEUF(1) = IBLF(JDAY)
KEUF(2) = IBLF(JTIM)
GO TO 900
C DC NOT PERMIT A ROW VALUE OF ZERO
IF (ITIME .EQ. 0) ITIME = 24
C WRITE OUTFLT ROW HEADER
IF (KUGC .NE. 0) CALL SCESS(* WRITING ROW HEADER FOR ROW *, ITIME)
IF (KPLG .NE. 0) CALL SCESS(* THIS IS MD FILE NC *, MDG)
ICK = MDPUT(MCC, ITIME, 0, KBUF)
IF (IOK .LT. 0) GC TO 540
40 NREP = 0
KOUT = 0
NREP = NREP + 1
C PICK UP COLUMN HEADERS
IF (MODC(I).NE.0) ICK = MDGET(AFIL, 0, NREP, IBUF)
IF (ICK .NE. 0) GC TO 50
C IF (IBUF(JTYF).NE.0) GO TO 40
ICK= NDFGET(AFIL,ISV,AREP,IBUF)
158 IF (ICK. NE.0) GC TO 40
159 ICAT(1)=IELF(LLAT)/100
160 ICAT(2)=IELF(LLON)/100
161 IF (ICAT(1).LT.MINLAT.OR.ICAT(1).GT.MAXLAT) GC TO 40
162 IF (ICAT(2).LT.MINLON.OR.ICAT(2).GT.MAXLON) GC TO 40
163 C SET USER MCC FLAG KEUF(9)
164 KBUF(4)=0
165 DC 45 K=1*13
166 I=LCCS(K)
167 IF (I.LT.0) GC TO 45.
168 KBUF(K)=IELF(I)
169 C 45 CONTINUE
170 C CHANGE TO TSL,CC, AND ADD Z10
171 IF (IELF(UEP).NE.MS) GC TO 40
172 IF (IELF(UDEW).NE.MS) KVELF(10)=IELF(UDEW)-IELF(UDEW)
173 C CHECK REASONAEBELNESS OF TOPOGRAPHY
174 MLAT=ICAT(1)
175 MLON=-ICAT(2)
176 CALL HRTCPG(MLAT,MLON,IEL,ICH)
177 Z=0
178 C GIVE SHIP FILE SPECIAL TREATMENT
179 IF (IEL.EQ.0.AND.ELV.EQ.-1) GC TO 46
180 IF (IAES(IEL-KVELF(4)).GT.1000) GC TO 40
181 Z=KBUF(8)
182 46 TSL=-0.01*FLCAT(KEUF(9))+Z=0.0065
183 KEUF(9)=100.0*TSL
184 C FIND Z10CC
185 IF (IBUF(JPRE).EQ.MS) GC TO 48
186 PSL=0.1*FLCAT(IBUF(JPRE))
187 PST=PSL+(1--0.00002266944*Z)**E.256
188 ALC=ALCG(1000./FST)
189 Z10C=Z-29.2858*TSL+ALG/(1+0.09519*ALG)
190 KBUF(14)=Z10
191 48 CONTINUE
192 C OUTPUT RECRC
193 NCUT=NOUT+1
194 KOUT(2)=ICAT(1)
195 KOUT(3)=ICAT(2)
196 KOUT(4)=KVELF(14)
197 KOUT(5)=KVELF(5)
198 KOUT(6)=KVELF(10)
199 KOUT(7)=KREP
200 KOUT(8)=NOUT
201 KOUT(1)=7
202 IF (KVELF. NE.0) CALL CUTFMT(KOUT)
203 ICK=MODPLT(MCC,ITIME,NCUT,KVELF)
204 IF (IOK.LT.10) GC TO 540
205 CC TO 40
206 50 CONTINUE
207 IF (NOUT.LT.10) GC TO 990
208 53 CALL SCST(* NC. CF REPCPTS WRITEN=* NOUT)
200 C    MDCO DATE TIME MDCO TO VASTEXT
201   IDCC(27)= (ICAY+ITIME) 16C
202   IDCC(30)=MC
203   IDCC(37)=ITIME
204   IF (LUC(F1).EQ.0) GC TO 65
205   CALL CWRITE(LUC, ITERN, IDCC)
206   GC TO 60
207   55 CALL TSAIO(3,1,1,1,1,IDCC)
208   60 CONTINUE
209 C    REWRITE RC8 Header WITH NO OF REF CRTS ADDED
210   KBUF(3)=NCRT
211   IOK=MDPUT(MDC, ITIME, 6, KBUF)
212   IF (IOK.NE.0) GC TO 540
213   RETURN
214  560 CALL SCST(" FILE NOT AVAILABLE FOR PROCESSING", 0)
215   RETURN
216  900 ITRY=ITRY +1
217   IDHR=ICHHR-100
218   IHR=IHR -1
219   IF (ITRY.LT.3) GC TO 10
220   CALL SCST(" INSUFFICIENT SURFACE DATA AVAILABLE", 0)
221   GC TO 33
222  540 CALL SCST(" CANNOT WRITE OUTPUT FILE", 0)
223   RETURN
224  960 CALL SCST(" CURREN MDFILE NOT DEFINED", 0)
225   RETURN
226  970 CALL SCST(" UNABLE TO OPEN INPUT FILE", 0)
227   RETURN
228  END
C--- DESCRIPTION OF 64-WORD GRID HEADERS
C
REAL*8 SUM*,SLMS*,FN
INTEGER IGH*,(64)
INTEGER IGN*,(6)
C-------GIVE TOTAL SIZE (WCRES) , # ROWS , # COLS . (IGSIZ=IGNR*,IGNC*)
EQUIVALENCE (IGSIZ,IGH*(1)),(IGNR*,IGH*(2)),(IGNC*,IGH*(3))
C-------YYCCD*, H-MMSS AND VALID-TIME (IF APPlicable) FOR GRIC
EQUIVALENCE (ICYAY*,IGH*(4)),(ITIM*,IGH*(5)),(IGTVM*,IGH*(6))
C-------DESCRIPTION OF GRIDDED VARIABLE (IN MT-FILE TERMS):
C------- NAME*, SCALE*, AND UNITS
EQUIVALENCE (ICVMA*,IGH*(7)),(ICVSCA*,IGH*(9)),(ICVUNI*,IGH*(5))
C------- DESCRIPTION OF VERTICAL LEVEL: VALUE*, SCALE*, AND UNITS
EQUIVALENCE (IGLEV1, IGCH(10)), (IGFSCA, IGCH(11)), (IGLUNI, IGCH(12))
C----- GRID - VARIABLE TYPE: 1 (TIME CIF) 2 (TIME AVG) 4 (LEVEL DIF)
C----- 8 (LEVEL AVG) OR ANY SUM OF THE FOREGOING
C----- EQUIVALENCE (IGTYPF, IGCH(13))
C----- FOLLOWING USED IF PARAMETER IS A VERTICAL (LEVEL) DIF OR AVG
C----- (SAME SCALE AS IGLEV1)
C----- EQUIVALENCE (IGLCIF, IGCH(14))
C----- FOLLOWING USED IF PARAMETER IS A TIME CIF OR AVG (HHMMSS)
C----- EQUIVALENCE (IGTCIF, IGCH(15))
C----- GRID ORIGIN, TYPE (I.E. TYPE OF PROJECTION)
C----- EQUIVALENCE (IGORG, IGCH(33)), (IGTYPF, IGCH(34))
C----- SEGMENTS (IGCMA, IGCH(35)), (IGCMB, IGCH(36))
C----- IGMLC, IGCH(37), IGMLC, IGCH(38) ALL HAVE 4 IMPLIED DEC. PLACES.
C----- LAT GOES FROM -900000 TO 900000, LON GOES FROM -1800000 TO 1800000
C----- TYPE 1 GRIDS ARE PSEUDOC-MERCATOR
C----- EQUIVALENCE (IGLAMA, IGCH(39)), (IGLOMX, IGCH(40)), (IGLAMN, IGCH(41)),
C----- (IGLCMA, IGCH(35)), (IGLCRM, IGCH(36))
C----- TYPE 2 GRIDS ARE POLAR-STEROGRAPHIC
C----- GIVE ROW # OF NORTH POLE, COL # OF N.P., CCL SPACING AT 60 DEG N.
C----- (DEG), LCALITUDE PARALLEL TO COLUMNS (DEG)
C----- EQUIVALENCE (IGPCLL, IGCH(35)), (IGPCLC, IGCH(36)), (IGFSCA, IGCH(41)),
C----- (IGFSCB, IGCH(42))
C----- INITIALLY OF USER AND PROJECT # UNDER WHICH GRID CREATED
C----- EQUIVALENCE (IGUSER, IGCH(41)), (IGFSCA, IGCH(42))
C----- CHARACTER IC SUPPLIED BY PROGRAM (ARBITRARY)
C----- EQUIVALENCE (IGID, IGCH(43))
C DATA MISS/ZE0208080C/*MCC/4/*NSIZE/2400/*NSMAX/595/
C DATA LBUF/33/Z40404040/, IBLNK/240404040/
C DATA SCL/0.0, 0.3, 0.5, 0.7, 1.0, 3.0, 3.0,
C X X T IC U, V IZ 2
C DATA MFILF, LLN, LEN/*VASTEXT */20, 100/
C DATA ICPL/1, 2, 3, 4, *T *, *T *, *T *, *CC */*LPR/*MSL */
C CALL SDEST(* BEGIN SFPCFG*, 0)
C IF (LUC(81), EGD.G), GC TO 1
C FOLLOWING FOR VAS APPLICATIONS
C PICK UP CONTEXT INFORMATION
C TERM=LUC(-20)
C CALL DOPEN(MFILF, LLN, LEN)
C CALL DREAD(LLN, ITREM, IDCC)
C GC TO 2
C FOLLOWING FOR TCVS APPLICATIONS
C 1 CALL TSNIC(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
C 2 MDNS=IDCC(36)
C MDRS=IDCC(37)
C MDNG=IDCC(38)
C MDNG=IKWP(*MCN*, 1, MDNG)
C NGRFS=IFP(2, 0)
C KFU=IKWP(*FLE*, 1, 0)
C CGS=CKWP(*ESS*, 1, 0)
IF (HDF:EQ(0,1CNS,0)+LT.0) GO TO 920

C GET AC. CF REPORTS FROM ROW HEADER
ICK=MOD(ICK,MCNS,MCPS,0,KELF)
ISYC=KPUF(1)
IHYS=KPUF(2)
IF (ICK.LT.0) CC TO 930
NRPT=KPUF(3)
NRPT=IKWF("RPT",1,NRPT)

CALL SFFST(* TCTAL AC. REPORTS = *NRPT)

C CHECK OPERATOR FORCED BOUNDARIES
LLNW=IDCC(25)
LLNW=IDCC(25)
LSE=IDCC(26)
LN=LLNW/1000
LS=LSE/1000
LW=MOD(LLNW,1000)
LE=MOD(LNW,1000)
L=IKWP("LAT",1,0)
IF (L.LE.0) L=W=L
L=IKWP("LAT",2,0)
IF (L.LE.0) LN=L
L=IKWP("LON",1,0)
IF (L.LE.0) LE=L
L=IKWP("LON",2,0)
IF (L.LE.0) LW=L
LINC=IKWF("LINC",1,0)
LINC=+1 FLCAT(LINC)

C USE IDATA AS FLAG IN EVENT CF NC DATA
IDATA=77777
IF=0
CHAR=CPF(1,*)
IF (CHAR.EQ.(*TSL *)) IP=3
IF (CHAR.EQ.(*DC *)) IP=4
IF (CHAR.EQ.(*U *)) IP=5
IF (CHAR.EQ.(*V *)) IP=6
IF (CHAR.EQ.(*TS *)) IF=7
IF (CHAR.EQ.(*Z100 *)) IF=8
IF (IP.EQ.0) CC TO 500

C SET OPTIONAL CROSS ERROR CHECK
SCL(IP)=CKWF("ERR",1,SCL(IF))

C INITIALIZE ANALYSIS FIELD
DO 10 L=1,NSIZE
IDL(L)=0
10 ITG(L)=0
ROWS=LH-LS

C AVCD SO HEMISPHERE PROBLEMS
LW=JABS(LW)
LE=JABS(LE)
IF (LW.GT.180) LW=LW-360
IF (LE.GT.180) LE=LE-360
IF (LW.LT.LE) LW=LW+360
COLS=LE-LE
FTS=(RCWS+1.):(CCWS+1.)
C ESTABLISH GRID INCREMENT IN 10THS OF DEGREES
FN=FLOAT(NSIZE)
IF (LINC.EQ.0) DIAC=SQRT(FTS/FN)
INC=DIAC 10.*+0.4*E
LAN=LN-10
LAS=LS 10
LCK=LW 10
LGE=LE 10

DINC=INC
RCWS=(LAN-LAS)/INC+1
CCWS=(LCK-LGE)/INC+1
CALL SCGET(*, NCWS, NCOLS, INC*, 0)
KOUT(1)=3
KOUT(2)=NCWS
KOUT(3)=CCCLS
KOUT(4)=INC
CALL OUTINT(KOUT)
NFTS=NROWS NCCLS
IF (LINC.EQ.0) GC TO 13
IF (NFTS.LE.NSIZE) GC TO 13
INC=INC+1
GC TO 11
13 CONTINUE
TLAT=LN
SLAT=LS
WLON=LW
ELON=LE
DINC=0.1 CINC
LF=IP
IF (LF.EQ.8)LF=2
C USE LP IN INSTANCES WHERE GSS AND DCC ARE INVOLVED
IF (LF.EQ.7)LF=3
C CHECK GUESS OPTION.. FILL FROM *VASTGSS*
IGGS=LIT(CGS)
C IF (CGS.EQ.*E**) GC TO 120
CALL SRGSS(IDC, ITG, NCCLS, NCWS, TLAT, WLON, CINCI, LF, MD) Y
DO 109 N=1,NFTS
109 ITCN(I)=ICL(N)
GC TO 106
120 CONTINUE
IF (CGS.EQ.*E**) GC TO 110
C EXPLOIT GRIDFILE GUESS OPTION
NGRF=ICGS/100
NCR=MOD(ICGS,100)
ICK=IGCRER(NGRF,IFIL1)
IF (ICK.EQ.0) GC TO 595
ICK=IGGET(NGRF, NCR, 2400, IDL, NR, NC, IGHD)
IF (NR.EQ.NCWS.OR.NC.EQ.NCOS) GC TO 590
IF (LF.EQ.2) GC TO 106
DC 105 N=1,NSIZE
208 ICL(N)=ICL(N)+10
210 ITC(N)=ICL(N)
211 105 CONTINUE
212 106 IF (I(KP*ACCAT,1,0)GE 0)GC TO 150
213 FCK=SDL(IF)-3.
214 INTO=FCK
215 IF (KEUG*AE=0)CALL SCEST (*TGSSCLT IS *ITCSS)
216 C ABOVE IS CROSS ERROR CHECK FOR GUESS FIELDS
217 C THIS CHECK IS PERFORMED ON 1ST ITERATION ONLY
218 110 NE=0
219 SUM=0.
220 SUMS=0.
221 DC 108 N=1,NPTS
222 108 FLD(N)=.01 FLCAT(ITC(N))
223 C NCTE=ITG IS USED TO SAVE ORIGINAL GUESS FOR ITERATION
224 DO 148 N=1,ARPT
225 IOK=MGET(MDNS,MDRS,N,KEUF)
226 IF (KEUF(IFC).EQ.MISS)CC TO 148
227 IF (IOK.LT.0)GO TO 148
228 LAT=KBUF(5)
229 LON=KBUF(6)
230 FLAT=FLAT(LAT):.0061
231 IF (FLAT.GT.TLAT.OR.FLAT.LT.SLAT)GC TO 148
232 FLAT=(FLAT-FLAT)/CINC+1.0
233 FLON=FLAT(LON):.0061
234 IF (FLON.GT.LON.OR.FLON.LT.ELON)GC TO 148
235 FLON=(FLON-FLON)/CINC+1.0
236 IF (KEUF(IFC).EQ.MISS)CC TO 148
237 TSL=.01 FLCAT(KEUF(5))
238 DC=.01 FLCAT(KLB(14))
239 240 Z=KBUF(1)
240 C ALL VARIABLES ARE IN TRUE UNITS
241 IF (IFC.GT.7)FCAT=TSL-.0061-Z
242 IF (IFC.GT.4)FCAT=CC
243 IF (IFC.GT.3)FCAT=TSL
244 IF (IFC.GT.3)FCAT=KBUF(14)
245 IF (CGS.NE.6)GO TO 330
246 IF (FCK.GE.555555)GC TO 330
247 C GET VALUE FROM GUESS GPC
248 CALL VALUE(LAN,LAN,LCL,LCL,LOE,INC,ICL,100,VAL)
249 IF (ABS(VAL-FCAT).LT.FCK)GO TO 330
250 ICAT=FCAT
251 IF (KEUG*AE=0)CALL SCEST (* REJECT DATA VALUE *IVAL)
252 KBUF(IFC)=MISS
253 IOK=MGET(MDNS,MDRS,N,KBUF)
254 GC TO 148
255 330 IDAT=FCAT.1CC.
256 NE=NE+1
257 C FILL BARNES ARRAYS
258 SUM=SUM+FCAT
259 SLS=SUM+FCAT*FDAT
260 DAS(NB)=FDAT
261 IREP(NE)=N
262 STLAT(NB)=FLAT
263 STLGC(NB)=FLGC
264 DA(NB)=FDAO
265 RW(NB)=FLATU
266 CL(NB)=FLCN
267 ICDA=0
268 IF (NB.LT.ROLE) GO TO 148
269 CALL SEDEST(* TCC MUCH DATA...EXITING TO ANALYSIS*,0)
270 GO TO 140
271 140 CONTINUE
272 140 IF (ICDATA.EQ.77777.AND.CGS.NE.(*G *))GO TO 400
273 IF (ICDATA.EQ.77777)C TO 150
274 C SET UP FOR BARNES ANALYSIS
275 IGB=0
276 IF (CGS.NE.* *)IGE=LIT(*GRG *)
277 ISCL=10000/INC
278 C SET BARNES SCALING
279 ISCL=IKWF(*SCL*1,ISCL)
280 C CALL SEDEST(* BEGIN BARNES WITH SCALING OF *,ISCL*)
281 CALL FEARN(TLAT,NRCLS,NCCLS,FLD,WT1,WT2,DAS,RLS,CL,
282 NE,INC,ISCL,IGE)
283 DC 142 N=1,NFNS
284 C AVOID NEGATIVE DEWPOINT DEPRESSION
285 IF (IP.EQ.4.AND.FLD(N).LT.0) FLD(N)=0.
286 142 IF (IKWF(*ECIT*1,0).NE.0) GO TO 150
287 C CHECK ON THE EDIT OPTION
288 IF (IKWF(*ECIT*1,0).NE.0) GO TO 150
289 FCK=999999
290 FN=NB
291 SUM=SUM/FA
292 SLMS=SUM/SFA
293 SD=CSQRT(SLMS-SLMS SUM)
294 C LIMIT SIZE OF STANDARD DEVIATION
295 IF (KELG.NE.0) CALL SEDEST(*DATA SD IS *,ISD*)
296 C LIMIT SIZE OF STANDARD DEVIATION
297 SD=AMINI(SD,SCL(IP))
298 MCUT=0
299 DC 130 N=1,NE
300 FLAT=STLAT(N)
301 FLON=STLCM(NN)
302 C GET VALUE FROM GRID
303 CALL VALGET(LAN,LAS,LCW,LCE,INC,ID1,100.,VAL)
304 IF (VAL.EQ.999999.) GO TO 130
305 NV=VAL
306 DAT =DA(NN)
307 DIF=DAT-VAL
308 C GRCSS ERROR CHECK
309 C
310 ADF=ADF(C,1)
311 IF (ADF.LT.SD) GO TO 130
312 ICK=MCGET(MCNS,MCES,IREP(NN),KBUF)
MOUT=MOUT+1
KRUF(IF+6)=MISS
ICK=MCPUT(KCNS,KCPSS,KEF(NM),KBUF)
IF (KALG,EG,C) GC TO 130
CALL ENKCE('*(1*IF*2F7.2,2F10.1)*',LBUF,NM,FLAT,FLCN,DAT,VAL)
130 CONTINUE
IF (NOUT.GT.0) GC TO 110
150 CONTINUE
C OUTPUT FIELD
ICVNAME=LMIT(CHAR)
IGVSCA=2
IGVUNI=IGUNI(IF)
IGLEV=LPF
IGLSCA=2
IGLUNI=1
IGVTYP=8
IGLDIR=1
IGCROG=0
IGTYPE=1
IGLAM=LAN 1000
IGLMN=LAS 1000
IGLCMX=LCW 1000
IGLCMN=LEC 1000
IGINCR=INC 1000
IGDAY=MOD(ISYC,100000)
ICTIME=IPMS
ICSIZE=NPTS
IGNR=NROWS
IGNL=NCOLS
ICK=IGOPEN(NGRFS,IFIL2)
IF (IOK,KE,C) GC TO 995
ICK=IGPUT(NGRFS,NGRNO,IDL,NROWS,NCOLS,IGHD,ISTAT)
CALL TMES('** ANALYSIS FILED AS GRID NO.5**,ISTAT)
IDO(C29)=NGRFS
ICOCX(28+LP)=ISTAT
IF (LUC(E1),EG,C) GC TO 280
CALL CWRITE(LLN,ITERM,IDCC)
CALL DCLOSE(LLN)
RETURN
280 CALL TSNIC(3.1.1,1.1,1,1,IDCC)
RETURN
400 CALL SCSEST('** NO DATA AVAILABLE FCR IMAGE** ,0)
RETURN
500 CALL SCSEST('** IMPFRCPER PARAMETER KEYED** ,0)
RETURN
700 CALL SCSEST('** (Z,T,C) GRID LLNW LLSE INC 10 G GE BSC**,0)
CALL SCSEST('** GRID IS GRID FILE AC. FCR SFC GRICS** ,0)
CALL SCSEST('** C=ASSUM (C=AC GUESS,C=LSE VASCSS,MM=GRID NO.**,0)
CALL SCSEST('** CE=GRCSS ERRGR OVERRIDE (** 10,DEG+100)** ,0)
CALL SCSEST('** ESC BARNES SCALE**,0)
RETURN
920 CALL SCSEST('** UNABLE TO OPEN MD DATA FILE NO.**,MDNS)
365 IF (CGS.EQ."G") GO TO 3
366 RETURN
367 930 CALL SCESM(AC, AC DATA AVAILABLE FOR HOUR, MDHS)
368 IF (CGS.EQ."G") GO TO 3
369 RETURN
370 990 CALL SCESM(" GUESS GRID NOT COMPATIBLE...", 0)
371 RETURN
372 995 CALL SCESM(" UNABLE TO OPEN GRID FILE...", 0)
373 1000 RETURN
374 END
SUBROUTINE MAIN

C ? * MODE 1
C ? EDIT VALUES IN MFILE FOR REPORT AT CURSOR LOCCATION
C ? HIT SPACE TO DELETE KEYIN "R END" TO TERMINATE
C ? MD LINKAGE THRUSWET AND VASTEXT FILE THUS RESTRICTED TO
C ? RETRIEVAL AND SURFACE MD FILES
C ?
C ? KEYIN: XRAVA <PARAM> <LEVEL>
C ? POSITIONAL PARAMETERS:
C ? PARAMETER KEYS FROM SCHEMA
C ? LEVEL PRESSURE LEVEL IF UNSPECIFIED ALL LEVELS DELETED
C ? WITH NO PARAMETERS ENTIRE REPORT IS DELETED
C ?
C ? * MODE 2
C ? EDIT SINGLE RETRIEVAL BY NUMBER
C ? KEYIN: XRAVA NRET,NN
C ? N IS NO. OF RETRIEVAL
C ?
C ? SEC/MCICAS LSERS MANUAL - CHAP12
DIRECTION MF(64),ICLT(300)
+IRMAX(40),IRMIN(40),IPMAX(40),IPMIN(40)
+ILAMIN(40),ILOMAX(40)
+ILAMAX(40),ILOMIN(40)
DIRECTION LIST(300),ISCL(300),LUO(300),LOC(300)
+IAN(20),MDHE(64),FLA(4),FLO(4),KOUT(10)
+INTEGR 4 CEF(20)
REAL*8 DLIT
CHARACTER*3 MFILE
CHARACTER*12 CLIT,ICFP,CFP
COMMON /DCC/IDCC(100)
COMMON /NAV,FLAT,FLCN,ZEN,LOC,SZEN,IL,IE,IRAS,IFIC,IPMS,JT,JD
COMMON /ICENT/IYCE,IPMS,KROWN,NSAT
COMMON /ANALS/KAAT,LTOP
COMMON /CPIT/NODE
COMMON /TIGHT/ITCL
COMMON /ENTRY/IAIT
COMMON /CIENT/YCCRD,YCCORD
COMMON /FLCNAV/NAVAIN,BETDCT,INAV,PTIME
DATA MFILE,LLN,LEN/"VASTEXT",20,100/
DATA IOUT /100 280808080/
DATA MISS/280808080/,NCCLS/56/
MDNC=LUC(E)
IWRP=LUC(1)
IFRP=LUC(-1)
C FORCE HIGH RESOLUTION SRCH
ITOL=1
IF (LUC(91),EG,0)GC TO 120
C READ VASTEXT CONTEXT FILE
ITRM=LUC(-20)
CALL DOPEN(CLIT(MFILE),LLN,LEN)
CALL DREAD(LLN, NTERM, IDCC)
C INITIALIZE NAVIGATION
IKAS=LLC(-11)
IPIC=LUC(-12)
CALL TVSAT(IFRM, IRAS, IPIC, IL, IE, JS, JD, JT)
JCTUS 1000CC+L
CALL NVINIT(FIETAIN, BDECT, INAV, FTIME)
GO TO 130
120 CALL TSKIC(1, 1, 1, 1, 1, 1, 1, IDCC)
NODE=IDCC(7)
INIT=1
130 CONTINUE
DC 1 K=36+41, 2
MDR=IDCC(K+1)
IF (MDNC. GE. IDCC(K)) GO TO 2
1 CONTINUE
GO TO 900
2 IF (MDOPEN(MNCO, 2). NE. 0) GO TO 900
IF (MDINFO(MNCO, MHC). NE. 0) GO TO 900
C CHECK FOR INDIVIDUAL DELETION
NRET=IKWP("NRET", 1, 0)
IF (NRET.GE.0) GO TO 99
IF (MDCH(E), NE. 0) ICK=MDPUP(MDNC+0, NRET, IOUT)
I0K=MDPUP(MDNC+MDR, NRET, IOUT)
IF (I0K.GE.0) GO TO 906
CALL SCAST(* PURGE SCOUND NO. *, NRET)
GO TO 110
99 CONTINUE
C READ ROW HEADER RECORD
ICK=MDGET(MDNC+MDR, 0, IOUT)
IF (I0K.LT.0) GO TO 902
MREC=IOUT(3)
C SAVE TCTAL TC DECREMENT AND UPDATE VASTTEXT
MREC=MREC
C READ TEST RECORD
M=0
101 M=M+1
IF (M.GT.100) GO TO 904
C READ KEYS
NKEYS=MDKES(MDNC-1, LIST, ISCL, ILA, LCCS)
C FIND LAT/LOC ADDRES IN KEYLIST
DC 11 N=1, NKEYS
110 IF (LIST(N). EQ. LIST("LAT"). N=1
111 IF (LIST(N). EQ. LIST("LON"). N=1
102 IF (LIST(N). EQ. LIST(*MOD *.)) NMOD=N
113 CONTINUE
104 IF (KBUG. NE. 0) CALL SCAST(* NO OF KEYS IS *, NKEYS)
CALL GETERM(1FRM,NF)
MAG=MFL(10)
IF(MAG.LT.6)MAG=6
CALL INITFL(1NRM,C)
NY=0
ICHRS=CFF(1,"")
IF (ICHRS.EQ."*)GC TO 10
C FALL THRU FOR SELECTIVE DELETION
ILEV=IFF(2,C)*10
C LEVELS IN NC FILES ARE NB 10
IF (ILEV.NE.0)GC TO 4
C SET UP TABLE TO DELETE ALL VALUES OF CHARACTER
K=0
DC 3 K=1,NKEYS
IF (ICHRS.NE.CLIT(LIST(N)))GC TO 3
K=K+1
IADD(K)=N
3 CONTINUE
IF (K.EQ.0)GC TO 520
KTOT=K
GC TO 8
4 CONTINUE
C CHECK ON SPECIFIC LEVEL AND CHARACTER
DO 5 N=1,NKEYS
IF (LIST(N).NE.LIT("P "))GO TO 5
I=LCCS(N)
IF (ILEV.NE.ICUT(I))GO TO 5
NS=N+1
GC TO 6
5 CONTINUE
GO TO 930
6 KTOT=1
DO 7 N=NS,NKEYS
IADD(KTOT)=LCCS(N)
7 CONTINUE
IF (ICHRS.EQ.CLIT(LIST(N)))GO TO 8
GO TO 520
8 IF (KBLG.NE.0)CALL SDEST("VARIABLE ADDRESS IS ",IADD(1))
10 CONTINUE
IF(ICURC(IRS,IFC,IRINC,IFINC).NE.0)GC TO 20
NX=NX+1
CALL WRRCX(IRS,IFC,IRINC,IFINC,2)
IRAS=IRS-IRINC/2
IFIC=IFC-IFINC/2
IRMIN(NX)=IRAS
IFMIN(NX)=IFIC
CALL FLOC
FLA(1)=FLAT
FLC(1)=FLCA
IPAS=IRS+IRINC/2
CALL FLOC
FLA(2)=FLAT
157  \text{FLC}(2) = \text{FLCA}
158  \text{IFC} = \text{IFC} + \text{IFINC} / 2
159  \text{IRMAX}(\text{NX}) = \text{IRAS}
160  \text{IFMAX}(\text{NX}) = \text{IFIC}
161  \text{CALL FLOC}
162  \text{FLA}(3) = \text{FLAT}
163  \text{FLG}(3) = \text{FLCN}
164  \text{IRAS} = \text{IRAS} - \text{IRINC} / 2
165  \text{CALL FLCC}
166  \text{FLA}(4) = \text{FLAT}
167  \text{FLG}(4) = \text{FLCN}
168  \text{FLMAX} = \text{SSSSS}
169  \text{FLAMIN} = \text{SSSSS}
170  \text{FLOMAX} = \text{SSSSS}
171  \text{FLOMIN} = \text{SSSSS}
172  \text{DC 13 N = 1, 4}
173  \text{IF (FLA(N) \text{GT} FLAMAX) FLAMAX = FLA(N)}
174  \text{IF (FLG(N) \text{GT} FLAMAX) FLAMAX = FLG(N)}
175  \text{IF (FLA(N) \text{LT} FLAMIN) FLAMIN = FLA(N)}
176  \text{IF (FLG(N) \text{LT} FLAMIN) FLAMIN = FLG(N)}
177  \text{CONTINUE}
178  \text{KOUT(1) = 4}
179  \text{KOUT(2) = FLAMAX 10}
180  \text{KOUT(3) = FLAMIN 10}
181  \text{KOUT(4) = FLGMAX 10}
182  \text{KOUT(5) = FLGMIN 10}
183  \text{IF (KEUG \text{NE} 0) CALL CUTINT(KOUT)}
184  \text{ILMAX(NX) = FLAMAX 10000}
185  \text{ILOMAX(NX) = FLGMAX 10000}
186  \text{ILAMIN(NX) = FLAMIN 10000}
187  \text{ILOMIN(NX) = FLGMIN 10000}
188  \text{NAV COMPLETE}
189  \text{CALL ENACPLT}
190  \text{IF (NX \text{NE} 0) GO TO 30}
191  \text{GO TO 10}
192  \text{GO TO 30}
193  \text{CALL SCDEST (*** ACTING TO DELETE *** 0)}
194  \text{CALL ENDPLT}
195  \text{RETURN}
196  \text{CALL SCDEST (** BEGINNING DELETIONS FROM FILE *** 0)}
197  \text{IF (LUC(81) \text{NE} 0) GO TO 31}
198  \text{SET UP AREA FOR IMAGE SPACE TC PLCT}
199  \text{CALL TVSAT(IFRM.005.335.LTCC.LEL.LEL.ISS.ID;IT)}
200  \text{CALL TVSAT(IFRM.495.335.LBOT.LEL.LEL.ISS.ID;IT)}
201  \text{IF (LTCC.LT.1) LTCC = 1}
202  \text{IF (LPTC.GT.ARCH) LPTC = ARCH}
203  \text{NROWS = LBOT - LTCC + 1}
204  \text{MPTS = NROW * ACOLS}
205  \text{DO 70 M = 1, MREC}
206  \text{IF (MCB(1) \text{NE} 0) ICK = MCGET(MCAO, C, M, ICUT)}
207  \text{ICK = MCGET(4CA0, MDR, M, ICUT)}
208  \text{C SKIP SOUNDINGS ALREADY DELETED}
I= (ICUT(MCCT).AE.0) GO TO 70

DC 40 K=1.NX

C DELETE EVERYONE WITHIN CURSOR
IF (ICUT(NLAT).LT.IAMIN(K).OR.ICUT(NLAT).GT.IAMAX(K)) GO TO 4
IF (ICUT(NLCN).LT.IOMIN(K).OR.ICUT(NLCN).GT.IOMAX(K)) GO TO 4

C THIS SCANNING MUST GC

C LOCATE RASTER AND FIXEL CF SOUNDING
FLAT=ICUT(NLAT)*.0001
FLON=ICUT(NLCN)*.0001
IF (LCU(81).EQ.0) GO TO 32

C REVERT TO -E CONVENTION
FLON=-FLON
CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLON,2,INAV,NRTAIN,EEDOT.0.)
      IL=FLIN+.0.E
IEL=FELE+.0.E
GO TO 33

C USE TCVS NAVIGATION RCLTINE
LATS=FLAT 100.
LONGS=FLON 100.
CALL SRCH(LATS,LCNGS,IM,IL,IE,NPTS,NROWS)
IF (IM.EQ.0) GO TO 40

C GUARD AGAINST OVERLAP IN SRCH TO AVOID DUPLICATION
IL=IL+LCIF-1
IF (IL.LT.LTIF.0.EG.LTIF) GO TO 40

C CALL SATTV(IFRM,IL,IE,IRAS,IPIC,JS,JE,UT)
IF (IRAS.LT.IRMIN(K).OR.IRAS.GT.IRMAX(K)) GO TO 40
IF (IPIC.LT.IPMIN(K).OR.IPIC.GT.IPMAX(K)) GC TO 40
CALL WRMAR(IRAS,IPIC,MAG,10)
KOUT(1)=4
KOUT(2)=IL
KOUT(3)=IE
KOUT(4)=ILALC(FLAT)
KOUT(5)=ILALC(FLON)
IF (KBUG.NE.0) CALL CUTFINT(KOUT)
GO TO 50

CONTINUE
GO TO 70

GO TO 55
ICUT(NMC)=5555555
GO TO 60

C REMOVE SELECTED VALUES
55 DC 58 N=1.KTCT
      I=IAEC(A)
      ICUT(I)="MISS"
58 CONTINUE

60 CONTINUE
      ICK=MDCPUT(MCCT,MDC,MCUT)
      IF (ICK.NE.0) GO TO 506
70 CONTINUE

100 CALL ENDPLT
    IF (LCU(81).NE.0) CALL DCLOSE(LLN)
110 CALL SEEEST(* ALL ECARE DELETING ...*,0)
261   RETURN
262   900 CALL SEST("CANNOT OPEN MFILE NO. "MENO)
263   RETURN
264   902 CALL SEST("CANNOT LOCATE ROW NO. "MDR)
265   RETURN
266   904 CALL SEST("TROUBLE READING DATA RECORD NO. "MP)
267   RETURN
268   906 CALL SEST("TROUBLE WRITING DATA RECORD NO. "MP)
269   RETURN
270   920 CALL SEST("REQUESTED PARAMETER DOES NOT EXIST IN SCHEMA")
271   RETURN
272   930 CALL SEST("CANNOT LOCATE DATA FOR LEVEL "ILEV)
273   RETURN
274   END
C ? VAS TOTAL=PRECIP+H2O + T(P) & W(P) RETRIEVAL (H,W,WCOLF)
C? KEYIN=VFZT <KEYWORDS>
C?
KEYWORDS (DEFAULTS IN PARENTHESES):
C?
AUTO=RUN WITHOUT IMAGE, GET ALL PARAMS FROM VASTEXT (0)
C?
ARA=NUMBER OF FIRST DIGITAL AREA FOR IMAGING (0)
C?
ARP=NUMBER OF SECOND DIGITAL AREA FOR IMAGING (0)
C?
*APA* IS FCP TOTAL PRECIPITABLE WATER VAPOR
C?
*ARP* IS FOR TOTAL-TOTALS
C?
NOTE ... IF *APA* IS ZERO, *ARP* IS FORCED TO BE ZERO
C?
BOX=A (SIZE IN FCV'S OF SCR BOX (11)) "EVEN-LE, TOP-DINDS"
C?
SRC=IL IE (LINE AND ELEMENT SPACING CF BOXES (BOX BOX))
C?
BEG=IL IE (FIRST LINE AND ELEMENT TO DEFINE AREA (VASTEXT))
C?
END=IL LE (LAST LINE AND ELEMENT TO DEFINE AREA (VASTEXT))
C?
GAM=100 (NORMAL CONDITIONING)
C?
GSS=G (CLIM) G (GRID; DEFAULT)
C?
SFC=1 (NO SURFACE ANALYSIS (0))
C?
TUP=1 (TERMINAL NUMBER (LOCAL))
C?
DUG=DIAGNOSTIC VALUES: 1=SCREEN, 2=PRINTER (0)
C?
The following pointers must be set with *SFWA*:
C?
MDR MPN NRET ... ALWAYS
C?
MDA MPD ... FOR GRID GUESS
C?
MDR, MDERS, NGFS, NGFS, ZGRID, TGRID, GGRID ... FOR SFC ANALYSIS.
C?
DIMENSION IRET(246), IRETC(246), IAIPES(6), IAIPES(66), IABUF(16)
C?
IAREA(2), IAREC(2)
C?
DIMENSION LELF(23), LSTT(17), VRM(13), VDAT(13), ELCB(12), COEF(9)
C?
DIMENSION Z(40), U(40), TC(40), W(40), GHS(20), GCS(11),
C?
TE(40), PST(15), TST(15), TST(15), DBR(15), SDF(15), STABIL(12)
C?
LCGR(20), LNE(2)
C?
EQUIVALENCE (IRET(17), IRETC(1))
C?
EQUIVALENCE TO IRET(ROULEN=1)
C?
ABOVE TO FACILITATE CHANGING LENGTH OF ROU HEADER
C?
INTEGER 2 IARRAY(11,11,11,4,2)
C?
CHARACTER 12 CGES, CKWP, CPLNK, CLET, CLET, CLETG
C?
COMMON/AREA/ICIP(64)
C?
COMMON/AMCS/F(40), T(40), W(40)
C?
COMMON/ATLLC/IECY, ILL, IEL, IER, MAG
C?
COMMON/DG/L/VM, VLM, VLM, VLM, VLM, VLM
C?
COMMON/OPUS/UEC
C?
COMMON/OC/ICC(100)
C?
COMMON/OPS/FTC(12,11,11), FTIC(2,11,11)
C?
COMMON/PS/FP(40,40), XIT(40,12), EFS(12), TEC(10), NCH, NFT, NFW
C?
COMMON/GAM/GAMEP, FRTEX, ERSFC, ERSFC
C?
COMMON/GC/GCSV(12), CSV(12), CSV(12), CSV(12)
C?
FOLLOWING IS PROGRAM VERSION DATE ... KEEP CURRENT!!!
IVER=84123
MCINIT=0
IF (LUC(-25),EG,1) MCINIT=LUC(-23)
CALL CALDAY(IVER,IVY,IVP,IVD,IVMC)
CALL ENKCODE("(132X,T1,"BEGIN " VTFZ : VERSION OF ",
I2.1X,94,12,1") AT INIT ",12/7",LEUF,IVE,IVMC,IVY,MCINIT)

C LOOK FOR DIGITAL AREA NUMBER
IARA=IKWP("*APA",1,0)
IF(IARA.GT.MAXARA) GC TO 1280
IAREA(1)=IARA
IARA=IKWP("*ARP",1,0)
IF(IARA.GT.MAXARA) GC TO 1280
IAREA(2)=IARA
IF(IAREA(1),EG,0) IAREA(2)=0
IAREAS=0
IF(IAREA(1),NE,0) IAREAS=IAREAS+1
IF(IAREA(2),NE,0) IAREAS=IAREAS+1
IARA=IAREA(1)

C CHECK FOR OVERRIDE OF TERMINAL NUMBER
IDCF=LUC(-20)
ITEM=IKWP("*TEP",1,1,1)
READ *VASTEXT* DOCUMENTATION RECORD & EXTRACT NEEDED INFO
CALL VRPTC(IRET,0,0)
JDAY=IODC(1)
JTIME=IODC(2)
MCAC=IODC(3)
MCAR=IODC(4)
MFR=IODC(4)

C CHECK FOR *AC-OUTPUT-MEDIUM*
IF(IARA,EG,6,AND,MERR,LE,4) GC TO 1300
115  LASRET=IRET(?)
116  CGES=CWIP(*WEX*.1,1,1)
117  IGES=6
118  IF(CGES.EQ.CELNK) GO TO 100
119  IF(CGES.EQ.CLETG) GO TO 100
120  IF(CGES.EQ.CLETC) IGES=1
121  100 CONTINUE
122  IDEF=7
123  MCSFC=IKWF(*SFC*,1,1,1)
124  NFSAV=CSFC
125  IDEF=11
126  NBXS=IKWF(*SFC*,1,1,1)
127  C NBXS SHOULD BE AN EVEN number FOR LAPGE-DETECTOR DATA
128  NDETR=0
129  IF(MOD(NBXS,2).NE.0) GO TO 110
130  NDETR=1
131  NBXS=NBXS-1
132  110 NBXS=MINC(NBXS,11)
133  IBOX=1
134  IF(NBXS.EQ.1) IBOX=0
135  IDEF=NBXS
136  INCRL=IKWF(*SFC*,1,1,1)
137  INCRE=IKWF(*SFC*,2,1,1)
138  C
139  LLINE=IKWF(*END*,1,1,1,1)
140  LELEM=IKWF(*END*,2,1,1,1)
141  ILINE=IKWF(*EGL*,1,1,1,1)
142  LELEM=IKWF(*EGL*,2,1,1,1)
143  C
144  LBEG=IKWF(*EGL*,1,1,1)
145  IF(LBEG.GT.1) CALL TGSET(2)
146  MLPI=ML+1
147  IFRM=LUC(-1)
148  INRAS=LUC(-11)
149  I*PIC=LUC(-12)
150  IGAM=IKWF(*IGAM*,1,1,1,100)
151  GAMRET=01 FLCAT(IGAM)
152  CAMSAV=GAMRET
153  CALL GETDAY(ICIPCS)
154  IFET(1)="CC(JCAY,.1000000)
155  IFET(2)=UTIME
156  IFET(3)=LASRET
157  IF(ILINE IELEM*.EQ.0) GO TO 140
158  IF(LUC(16).EQ.0) GO TO 170
159  IF(IKWF(*ALT*,1,1,1,.NE.0) GO TO 170
160  CALL GETFRM(IFRM,ICIR)
161  JSAT=ICIP(1)
162  JDATE=ICIP(2)
163  CC TO 160
164  140 CALL TVSAT(IFRM,INRAS,INPIC,ILINE,IELEM,JSAT,JDATE,UTIME)
165  C NEED "OLESTYLE" JCAY FOR NAVIGATION WITHIN VASCAT
166  160 JCAY=JSAT 100000+JDATE
157 170 IF((LLINE LELEM).NE.0) GC TO 150
158  LLINE=LLINE
159  LELEM=IELEM
160  150 CONTINUE
161  151 VCAT(1)=-1.
162  152 I LINES=IL IN E
163  153 IELEMS=IELEM
164  C VSACEAT CHANGES ARGUMENTS *ILINES* AND *IELEMS*
165  CALL VSACEAT(ILINES,IELEMS,VCAT)
166  154 VDAT(1)=VMISC
167  155 JSAT=IDIR(2)
168  156 ICAT=ISATNAV(I SAT)
169  157 JDATE=IDIR(4)
170  158 JDAY=JSAT+100000+JDATE
171  159 JTIME=IDIR(5)
172  160 ILCES=IDIR(12)
173  161 IERES=IDIR(13)
174  C SET DEFAULT TO SQUARE SAMPLES ... I.E., EQUATE RESOLUTIONS
175  162 INC=ILERES/IERES
176  163 INCIL=ILERES INCPL
177  164 IF(INCIL.EQ.0) INCIL=1
178  165 INCIE=IERES INCPE-INC
179  166 IF(INCIE.EQ.0) INCIE=1
180  C DEFINE AREA PARAMETERS (NEEDED EVEN IF AREA NOT TO BE WRITTEN)
181  167 INCAL=ILERES
182  168 INCEA=IERES INC
183  169 NFXT=NBXS/2
184  170 I L I N A=ILINE-1 NBXT+INCLA
185  171 IELEA=IELEM-1 NBXT+INCEA
186  172 LLINA=LLINE+1 NBXS+INCLA
187  173 LELEA=LELEM+1 NBXS+INCEA
188  C EMPRICAL ADJUSTMENTS!
189  174 I L I N A=ILINE-1
190  175 LLINA=LLINA-1
191  176 MLIN=(LLINA-ILINA)/INCLA+1
192  177 MELE=(IELEA-IELEA)/INCEA+1
193  178 KBXS=KELEMS*KELEMS
194  179 KBOXES=KLE/SKBOXES
195  180 IF((MOD(KELE+KELEMS),NE.0)) KBOXES=KBOXES+1
196  181 IF((NARA*EQ.0)) GC TO 200
197  C
198  182 IF((MELE*GT.660)) GC TO 1320
199  183 GC 190 IA=1+IAFEAS
200  191 KARA=IAREA(IA)
201  192 CALL ARASIZ(KARA,MLIN,MELE)
202  193 CALL EAAREA(KARA,JSAT,JDATE,JTIME,ILINA,IELEA,INCLA,INCEA,IEAMES)
203  194 CALL ORNA(KARA)
204  195 IAREA(IA)=0
205  196 CONTINUE
206  197 IA=IA+1
207  198 IACAP=INCPL-NBXS
208  199 KLINE=KLINES
200 CONTINUE
214 C
215 C BEGIN MAIN LINE LOC2
216 DC 126G IL=ILINE,LLINE,INCIL
217 LASL1N=-1
218 KECY=0
219 C BEGIN MAIN ELEMENT LOC2
220 RC 118G IE=IELE,KELE,INCIE
221 KEC=KEC+1
222 IF(LFUG.EG.0) GC TC 220
223 C BEGIN ENCORE("BEGIN LINE","IL","IL","ECX","IL","KBCX")
224 220 LAELE=-1
225 C INITIALIZE BOX PARAMETERS
226 DC 240 KLIN=1,KLINES
227 DC 240 KELE=1,KELEMS
228 DC 230 I=1,13
229 230 TDAE(I,KELE,KLIN)=VWISG
230 IF(KARA.EG.0) GC TC 240
231 GC 235 KARA=1,ARREAS
232 TARRAY(KELE,KLINES,KBCX,KARA)=MAXVW
233 235 CONTINUE
234 240 CONTINUE
235 C
236 C ACQUIRE VAS DATA FOR BOX
237 MSAM=0
238 SELEV=0.
239 LSTYPE=0
240 DC 330 KLIN=1,KLINES
241 DC 320 KELE=1,KELEMS
242 DC 260 I=1,12
243 VDAT(I)=VWISG
244 260 NSFIN(I)=1
245 IILINES=IL
246 IELE=IE
247 C IELE=IE
248 IF(ABS(VLATS).GE.50.) GC TC 320
249 IF(ABS(VZFR).GT.0.) GC TC 320
250 IF(LELEV.EG.500000) GC TC 320
251 IF(VDAT(I).LT.0.) GC TC 320
252 DC 300 I=1,12
253 IF(VDAT(I).LT.180.CO.VDAT(I).GT.330.) VDAT(I)=VWISG
254 IF(VDAT(I).EC.WTSP) GC TC 300
255 TCAT(I)KELE,KLIN)=VCAT(I)
256 C SAVE REPRESENTATIVE SFM BLOCK (ASSUME SFM BUDGET INVARIANT)
257 IF(ISFIN(I).NE.0) NSFIN(I)=ISFIN(I)
258 IF(I.EQ.8) GC TC 300
259 MSAM=MSAM+1
260 SELEV=SELEV+FLCAT(LELEV)
IF (ICHAR .LE. 0) ICHAR = 1
LSTYPE = LSTYPE + ICHAR
330 CONTINUE
340 CONTINUE
350 CONTINUE
360 CONTINUE
IF (YSAM.EQ.0) GO TO 1180
370 C DETERMINE DOMINANT SURFACE TYPE; SET *NO-SURFACE* FLAG OVER WATER
380 NSAM = NSAM / 2
390 NCSCF = NFSAV
400 C DATA ACQUISITION COMPLETE
410 C
420 C DETERMINE GUESS
430 CALL GESFRC (IGES, NCSCF, MCENG)
440 IF (IGES(1).LE.0) GO TO 1180
450 C
460 IF (LSTYPE .LT. NSAM) NCSCF = 1
470 PSTA = IPSTA
480 TSTA = C1 * FLCAT (JSCF)
490 TDSTA = C1 * FLCAT (ICSF)
500 CC = TSTA - TDSTA
510 CALL WMX (FSTA, TSTA, CC, WSTA, 1)
520 C FIND *IS*, FIRST LEVEL BELOW SURFACE (PRESSURE)
530 IS = NL
540 DC 340 J = 1 - 20
550 I = NLF1 - J
560 DP = F(I) - PSTA
570 IF (DP .GE. 0.) GO TO 340
580 IS = I + 1
590 GO TO 360
600 CONTINUE
610 340 CONTINUE
620 360 CONTINUE
630 IS = MIN (IS, AL)
640 ILC = 35
650 IF (IGES .EQ. 1) ILC = 31
660 IF (ILC .GE. IS) GO TO 400
670 CTS = TSTA - T(IS)
680 CTDY = CTS / (F(IS) - F(ILC))
690 DC 380 I = ILC + IS
700 380 T(I) = T(I) + CTDY - (F(I) - F(ILC))
710 400 CONTINUE
720 420 W(I) = W(IS)
730 DC 440 I = 1 - 40
740 T(I) = T(IS)
750 420 W(I) = W(IS)
760 DC 440 I = 1 - 40
770 W(I) = W(I) * WSTA / W(IS)
780 WSATC = WSAT (F(I) * T(I))
790 W(I) = WMAX(W(I) * C * 0.02)
800 W(I) = AVMIN (W(I), WSATC)
810 LG(I) = W(I)
820 TC(I) = T(I)
830 440 CONTINUE
IF (LEVF, F, E) "C TO 46"
CALL ENKCEE(*("PRESSURE", 15F7.1/), LEUF, F(26))
CALL ENKCEE(*("GUESS TEPF", 15F7.1/), LEUF, T(26))
CALL ENKCEE(*("GUESS WMP", 15F7.1/), LEUF, W(26))
CALL ENKCEE(*("21000 =", 18, E), "TSFC =", 18, E), "TSFC =", 18, E)
LEUF, IZ10, ITSFC, ICSCF)
460 CONTINUE

C GENERATE TEG'S FOR RETRIEVAL
IF (NOSFC, E, C) TSTA=AMIN1(TSTA, 278,)
CALL FILTER(VDAT, NSPIN, ERW, NSAM, TSTA, LCETR)
IF (NSAM, E, C) GC TO 1180
ICOUNT=0
DC 480 I=1, 10
IF (VDAT(I), E, CMISEC) GC TO 480
ICOUNT=ICOUNT+1
480 CONTINUE
IF (ICOUNT, LT, 7) GC TO 1180
TSTIM=VSTINT(VDAT, E, I, JACT)
IF (LEVF, E, C) CALL ENKCEE(*("VSTINT ACTION =", 12, E), LEUF, JACT)
IF (JACT, E, C) GC TO 1180
EFTSFC=1.0
ERWSFC=0.1
ERMSAV=ERM(S)
ERM(8)=1000.
IF (NOSFC, E, C) GC TO 485
TSTA=2. VDAT(8) - VDAT(7)
WSAT=0.005 (PSTA, TSTA)
TCSTA=TSTINT(PSTA, TSTA, WSAT)
EFTSFC=4.0
ERWSFC=0.25
ERM(8)=ERMSAV
485 CONTINUE

C PREPARE GUESS INFC FOR MCFILE
TCS(I)=TSTA
DC 1000 I=2, 20
J=LMD(I)
1000 TCS(I)=T(J)
DCS(I)=TCSTA
DC 1020 I=2, 11
J=LMD(I)
DFT=CFWPT(F(.E), T(J), 4(J))
1020 DCS(I)=DFT

C MAKE SOUNCING FOR ECX
IF (LEVF, E, C) CALL ENKCEE(*("CALLING GVTWP, IS =", 13, E), LEUF, IS)
GAMRET=GAWSAV
CALL GVTWR(VDAT, E, TC, TSKIN, TSTA, WSAT, IS, ISET, IFAIL)
IF (IFAIL, E, C) GC TO 1180
DC 1030 I=1, AL
TC(I) = T(I)
1050 WG(I) = (I)
367 GAMSAT = 0.1** GAMSAV
368 CALL GYTLP(I*V0AT,E**T0TT**TSKIN*TSTA*WSTA*IS*ISAT*IFAIL)
369 IF(IFAIL.NE.0) CC TC 1100
370 IF(GAMSAT.LT.5) GC TC 490
371 C
372 C SETAI ON THE ADDITION AL CUTFRT PARAMETERS
373 CALL FRECA(F*U.U.I.S)
374 DC 1040 I=2N**40
375 TC(I) = T(EWFT(F(I)**T(I)**U(I))
376 1040 CONTINUE
377 TST = T(37) + T(E(27)-2) T(31)
378 CALL HTV(Z*IS)
379 C SET UP FC FOR STABILITY CALCULATION
380 NB = IS
381 IF(FSTA.LT.F(NB)) NB = NB-1
382 DC 1060 I=2M+15
383 FST(I) = F(NF)
384 TST(I) = T(NF)
385 TST(I) = T(NF)
386 1060 NB = NB-1
387 FST(I) = FSTA
388 TST(I) = 01 ITSFC
389 TST(I) = 01 ITSFC
390 IF(LBUG.NE.0) CALL EJAOCE("STAB. PRESS ", 10F8.2) , LBUF,FST(1)
391 CALL SDAAN((C+1.5FST,TST,TST,TST,CIR,SPD,STAEB)
392 LIFT = IRPLNC(STAEB(I):100)
393 IF(LBUF.EQ.0) GC TC 1080
394 CALL EJAOCE("TAB. PRESSURE ", 15F7.0) , LBUF,F(26)
395 CALL EJAOCE("TAB. PROFILE ", 15F7.1) , LBUF,T(26)
396 CALL EJAOCE("TAB. HEIGHT ", 15F7.0) , LBUF,Z(26)
397 CALL EJAOCE("TAB. MIX RATIO ", 15F7.3) , LBUF,2(26)
398 CALL EJAOCE("T11","TCT =","F7.2") , LBUF,TCT)
400 1080 CONTINUE
401 C PREPARE CUTPLT EUFFER
402 DC 1100 I=4+NSIZE
403 1100 IRET(I)=MISC
404 IRET(4)=LLC(-17)
405 IRET(5)=IVER
406 IRET(6)=INCR
407 IRET(7)=INCRE
408 ICD = MOD(IDC, ICC(3)+1000)
409 IRET(8)=ICCC(3)+ICG+1000
410 IRET(9)=ISCC(6)
411 DC 1120 K=1**5
412 1120 IRET(F+K)=ISCC(K)
413 IRET(15)=0
414 IRET(16)=ICPRES
415 C FILL IN USER MOD FLAG
416 IRET(I)=0
IRET(D4) = LASFRET + 1
IRET(D5) = IRECLNF(VLAT 16000.)
IRET(D6) = IRECLNF(VLCN 10000.)
IRET(D7) = IRECC(2)
IRET(D8) = ASAP
CC 1140 K=1:12
IF(VDAT(N).EC.WISEC) GM TO 1140
IRET(D9) = IRCUAC(VCATA(N) 100.)
IRET(D10) = IRCUAC(ERMN(N) 10000.)
1140 CONTINUE
IRET(D11) = IRCUAC(UI(IS) 1000.)
IRET(D12) = IRCUAC(TSKIA 100.)
IRET(D13) = IRCUAC(S2EN 100.)
IRET(D14) = IRCUAC(WZEN 100.)
IF(FSTAEQ.EQ.0.) IRETD(41) = IRCUAC(TCT 100.)
IRET(D47) = LIFT
IRET(D50) = 21
C STORE DATA BY LEVEL
LS=NLFP1-F
LS=MAXC(LS,3)
CC 1160 K=2:20
L=LMD(K)
*=(K-1)*
IRET(D+7) = Z(L)
IRET(D+1) = LCHP(K)
IRET(D+2) = F(L).10.
IF(FLV.GT.FSTA) CC TO 1160
C BELOW SURFACE, STORE HEIGHTS ONLY
IRET(D+3) = IRCUAC(T(L) 100.)
IRET(D+6) = IRCUAC(TGS(K) 100.)
IF(K.GT.11) CC TO 1160
IRET(D+4) = IRCUAC(TC(L) 100.)
IRET(D+9) = IRCUAC(CGS(K) 100.)
1160 CONTINUE
C ADD SURFACE VALUES
IRET(D1) = LCHP(1)
IRET(D2) = IPSTA 10
IRET(D3) = ITSFQ
IRET(D4) = ICSCF
IRET(D5) = IELEV
IRET(D6) = IRCUAC(TGS(1) 100.)
IRET(D7) = IRCUAC(CGS(1) 100.)
C MD OUTPUT BUFFER COMPLETE
LASRET=LASRET + 1
IDCC(100) = LASRET
CALL VRIC(J,IRFET,LASFET,1)
C 450 IF(APA.EQ.0.) CC TO 1180
C GET NEW SFCV T-F-W AND TOTAL-TCTALS FOR IMAGE
NCHT=NCH+2
NNEW=NFW+1
NFTT=NFT+NFW
CC 500 KLIN=1*KLINFS
CC 540 KELE=1*KELEFS
IF1=IFLAG(1*KELE*KLIN)
IF2=IFLAG(2*KELE*KLIN)
ICHECK=0
IF(IF1.EQ.1.AND.IF2.EQ.0) ICHECK=10
CC 502 I=1+1?
DO 504 I=1+NCH
CTE(I)=VCAT(I)-TEC(I)
LTE(1*NCH)=KSTA-WG(IS)
LTE(NCH+2)=TG(IS)-TSTA
CC 506 I=1+NCH
X=ABS(CTE(I))
IF(X.GT.1) CTR(I)=0.
IF(ICHECK.EQ.1) GO TO 506
IF(I.GT.2.AND.I.LT.9) CTR(I)=0
506 CTE(I)=CTE(I)/ERS(I)
DC 510 J=1+1
SUP=0.
CC 508 K=1+NCH
SLM=SUM*XIT(J,K)*CTE(K)
CC 514 I=21+IS
SUP=0.
DO 512 J=1+1
SLM=SUM*XIT(J,K)*CTE(K)
FHI(J)=W(I)=WC(I)+CF1*SUM
CALL PRECW(F*1,L*15)
URET=U(IS)
URET=CF1*(URET**12)
IURET=IPCLAC(URET**20)
TARRAY(KELE,KLIN,KECX+1)=IURET
IF(IFACEA2.EQ.0) GO TO 540
IF(PSTA.LT.EC.C) GO TO 540
CC 518 T=21+IS
SUP=0.
DO 512 J=1+1
SLM=SUM*XIT(J,K)*CTE(K)
FHI(J)=W(I)=WC(I)+CF1*SUM
CC 510 J=1+1
SLM=SUM*XIT(J,K)*CTE(K)
T(I)=T(I)-SLM
CC 510 J=1+1
T(I)=T(I)-SLM
CC 510 J=1+1
T(I)=T(I)-SLM
CC 510 J=1+1
T(I)=T(I)-SLM
CC 510 J=1+1
T(I)=T(I)-SLM
CC 510 J=1+1
T(I)=T(I)-SLM
C IF(NARA.EQ.0) GO TO 1265
C OUTPUT TO DIGITAL AREA, ONE LINE AT A TIME
DC 1250 L=1,NAREAS
KARA=IAREA(L)
KREC=IAREC(L)
DC 1260 J=1,KLINEA
IA=0
DC 1260 K=1,KREC+1
DC 1260 I=1,KELEMS
IA=IA+1
IARET(I)=IARRAY(I,J,K,L)
1260 CONTINUE
CALL PACK(MELE,IARET,IAELF)
CALL WRITE(KAPA,KREC,IAELF)
KREC=KREC+1
1260 CONTINUE
IF(IAGAF.LE.0) GO TO 1265
C FILL CAPS IN IMAGE WHEN BOX LINE-SPACING .GT. BOX SIZE
DC 1230 M=1,MELE
1230 IARET(M)=MAXLV
CALL PACK(MELE,IARET,IASBF)
DC 1240 I=1,IAGAF
CALL WRITE(KAPA,KREC,IAELF)
KREC=KREC+1
1240 CONTINUE
1245 IAREC(L)=KREC
1250 CONTINUE
C
1255 CALL EDEST(*FINISHED LINE *,IL)
1260 CONTINUE
DCCC(100)=IASCRT
IRET(3)=IASCRT
CALL VRTIC(IRET,G+1)
IF(NARA.EQ.0) GO TO 1270
DC 1265 L=1,NAREAS
KAPA=IAREA(L)
CALL CLOSAC(KAPA)
1265 CONTINUE
1270 CALL EDEST(* - ALL DCMK ----- *,O)
RETURN
1280 CALL EDEST(*INVALID AREA NUMBER *,NARA)
GO TO 1340
1300 CALL EDEST(*NC CLUTFLT MEDIUM (AREA OR ME) SPECIFIED * * * *,O)
GO TO 1340
1320 CALL EDEST(*IMAGE TOO WIDE - MAXELE IS 660, FSIZ = *,MELE)
1340 CALL ACCEPT()?
RETURN
END
C ? PLOT ALL VALUES IN MC FILE (VAS RETRIEVALS)
C ? NC LINKAGE THRU VASTEXT OR KEYWORD
C ? KEYWORD: PLVA <FAR> <LEV> KEYWORDS
C ? FAR MAY BE ANY PARAMETER IN THE MC SCHEMA
C ? LEV MAY BE *SFC*, MB LEVEL, OR *DIFF* FOR TBB DIFFERENCES
C ? KEYWORDS:
   *LAT* MIN AND MAX LATS
   *LCN* MIN AND MAX LONGITUDES
   *LEV2* 2ND LEVEL FOR THICKNESS (LEV-LEV2)
   *MCNR* RETRIEVAL MC NC
   *MCRR* RETRIEVAL MC ROW NC
   *SIZE* SIZE OF PLOTTED DIGIT
   *CCLR* COLOR OF PLOTTED DIGIT
   *ALL* PLOT EVERYONE...EVEN THE REJECTED VALUES
   *TYPE* PLOT ONLY INDICATED TYPE (21,22, OR 23)
   *GRA* PLOT ON SPECIFIED GRAPHICS FRAME

DIMENSION SATFCS(2)
DIMENSION FF(64),LEUF(33),IOUT(300)
ILIAMIN(40),ILCIAN(40)
ILIAMY(40),ILCMAX(40)
DIMENSION LIST(400),ISCL(400),IUN(400),LOC(400)
DIMENSION KLCT(10),KCCHC(64)
CHARACTER 12 CCNP,LCHR,CPP,CLEV
CHARACTER & FFIL
CHARACTER & CLIT
COMMON /CC/IDCC(112)
COMMON /NAV/FLAT,FLCN,ZENLCC,SEAM,IL,IE,IRAS,IFIC,IFMS,JT,JC
COMMON /IDENT/IMC,IMHS,CMW,MNSAT
COMMON /ANALS/CANAL,LTOP
COMMON /CRED/ACCE
COMMON /TICT/ITCL
COMMON /ENTRF/INIT
COMMON/ORIENT/YCCL,CCRD,CCCCRD
DATA SATFCS/135,7E-/
DATA CCST/.17365/
DATA LEUF/33:Z409404040/
DATA MFIL,LLA,LEN/"VASTEXT *,20,100/
DATA MISS/28C804080/
DATA *CCS/5E/
C CHECK DEELC CFTION
IHEL=IKWF(*HELP,*1,0)
IF (IHEL,NE,0)GC TC 800
KEEBUG=IKWP(*ELG,*1,0)
ITYPE=IKWP (*TYHP*1,C)
ITRP=LUC (-F)
IRMP=IKWP (*GRA*1,IRMP)
IFRM=LUC (-1)
IF = (LUC(81).*EC.*EC) GO TO 110
C READ VASTEXT CONTEXT FILE
ITERM=LUC (-2C)
CALL DCPEN (*MTFL*LLN,LEN)
CALL DREAD (LLN,ITERM,IDOCC)
CALL DCLOSE(LLN)
C INITIALIZE NAVIGATION
IRAS=LUC (-11)
IFIC=LUC (-12)
CALL TVSAT (IFRM,IRAS,IFIC,IL,IE,ISS,JD,IT)
JC=ISS 100000+JC
CALL MVINIT (BETAIN,BETDOT,INAV,PTIME)
GO TO 120
110 CALL TSNIC (1,1,1,1,1,1,1,IDOCC)
INIT=1
NODE=IDOCC (7)
C SET UP AREA FOR IMAGE SPACE TO PLCT
CALL TVSAT (IFRM,005,0,335,LTCP,LELE,ISS,IC,IT)
CALL TVSAT (IFRM,495,0,335,LTCP,LELE,ISS,IC,IT)
IF (LTCP.LT.1) LTCP=1
IF (LBCT.GT.RRCWM) LBCT=RRCWM
NROWS=LBCT-LTCP+1
J=1
M=1
NPTS=RRCWM ACCLS
120 CONTINUE
ISS=0
IF (LUC(81).*EC.*EC) GO TO 125
JD=IDOCC (1)
ISS=ISATNY (JD)
SATLON=-SATFLCS (ISS)
C CHECK OPERATOR FORCED BOUNCERIES
125 LLNW=IDOCC (25)
LLSE=IDOCC (26)
LAN=LLNW/1000
LAS=LLSE/1000
LCW=MOD (LLNW,1000)
LCE=MOD (LLSE,1000)
L=IKWP (*LAT*,1,C)
IF (L.*NE.*0) LAS=L
L=IKWP (*LAT*,2,C)
IF (L.*NE.*0) LAN=L
L=IKWP (*LON*,1,C)
IF (L.*NE.*0) LCE=L
L=IKWP (*LON*,2,C)
IF (L.*NE.*0) LCW=L
LCW=IAES (LCW)
LCE=IAES (LCE)
IF (LC1.GT.180)LC1=LC1-360
IF (LC1.GT.180)LC1=LC1-360
IF (LC1.LT.LCE)LC1=LC1+360
LC1=0
IF (1ABS(LCW-LCE).GT.180)LCX=1
IF (LCX.NE.0.AND.LCW.LT.0)LCW=LCW+360
MCNC=IKWP(*MCNC+1,ICCC(40))
MCR=IKWP(*MCR+1,-1)
IF (MCR.LT.0)MCR=ICCC(41)
IF (MCDFC(MCNC,MCTR).NE.0)GO TO 900
IF (MCCOPEN(MCNC,2).NE.0)GO TO 900
C HAVE TO CALL MDOOPEN TO LOAD MDGET...
C READ ROW HEADER RECORD
ICK=MDGET(MCNC,MCR,0,ICUT)
KOUT(1)=3
KOUT(2)=ICLT(1)
KOUT(3)=ICLT(3)
KOUT(4)=ICLT(16)
IF (KBUG.NE.0)CALL CUTINT(KOUT)
IF (ICK.LT.0)GO TO 902
C READ TEST RECORD

125  *+=1
126  40  *+=M+1
127  IF (MCNC(8).NE.0)ICK=MDGET(MCNC,MCTR,0,ICUT)
128  MMXX=MDHC(5)
129  IF (KBUG.NE.0)CALL SDEST(*,MMAX IS *,MMAX)
130  C MMAX IS MAX POSSIBLE DATA ENTRIES USED TO AVOID
131  C INFINITE LOOP AT SN 80
132  C ICK=MDGET(MCNC,MCR,0,ICUT)
133  IF (M.NE.1)GO TO 904
134  IF (ICK.LT.0)GO TO 40
135  IF (KBUG.NE.0)CALL SDEST(*,OPERATING WITH RCW=*,MDF)
136  C READ IN KEYS
137  KEY=MCKEYS(MCNC-1,LIST,ISCL,IUN,LCCS)
138  IF (KEYC.NE.0)CALL SDEST(* NO OF KEYS IS *,NKEYS)
139  CALL GETFRM(IFRM,MF)
140  MAG=MF(10)
141  IF (MAG.LT.6)MAG=6
142  MAG=IKWP(*SIZE+1,MAG)
143  CALL INITPL(1,ICUT,C)
144  CCHR=CPP(1,*2,*)
145  LCHR=CCHR
146  IF (LCHR.EQ.*WIN*)CCHR=*SPC*
147  ISUE=0
148  IF (CCHR.NE.(*TEEF*))GO TO 20
149  ISUB=5
150  CCHR=(*T *)
151  20 IF (CCHR.NE.(*EEEF*))GO TO 30
152  ISUB=5
153  CCHR=(*TD *)
154  30 CONTINUE
155  C CHECK FOR SELECTIVE PRESSURE
ILEV = IFF (E.C) 10
CLEV = CFF (E.C) 10

C  IF (KBC.G.E.C) CALL SCESS (* LEVEL IS * , ILEV)

C  SET DEFAULT TO ECO Z

C  IF (CCHR.E.G.2.0.AND.ILEV.E.G.0.AND.CLEV.E.G.1.0) ILEV = 5000

C  LEVELS IN MC FILES ARE ME 10

C  CHECK FOR THICKNESS CALCULATION

C  ILE = IKNF (* ELEV * 1.0) * 10

C  CHECK FOR TEE DIFF

C  IF (CLEV.E.G.0.IIFF *) ISSUE = 13

C  FIX WHERE LAT/LOC INFO IS

C  DO 18 * = 1.MEY

C  IF (LIST(E).E.G.LIT(*MCN 0.0)) MCD = N

C  IF (LIST(E).E.G.LIT(*NMA 0.0)) NCA = N

C  IF (LIST(E).E.G.LIT(*MREC 0.0)) NR = N

C  IF (LIST(E).E.G.LIT(*LAT 0.0)) NLAT = N

C  IF (LIST(E).E.G.LIT(*LON 0.0)) NLO = N

C  IF (LIST(E).E.G.LIT(*RT 0.0)) NRT = N

C  SAVE REPRESENTATIVE ENTRIES FOR MCCA DATA CASE

C  IF (CLIT(LIST(E)).E.G.CCHR) NADD = N

18 CONTINUE

ISC = 10. * ISCL (NADD)
MREC = 500

IF (NCA.NE.0) MREC = IUCUT (NCA)
MREC = IKNF (*LAST 0.0, 1.0, MREC)

IF (KEFC.E.G.C) CALL SCESS (* NO. CF REPORTS IS *, MREC)
IALL = IKNF (*ALL 0.0, 1.0)

19 CONTINUE

NS = 1
M = 1

15 CONTINUE

IF (MCRC.E.G.C) GO TO 115

C  READ CCLMLA HEADER FOR RAOF

ICK = MCGET (MCRC, G.MM, ICUT)

115 CONTINUE

ICK = MCGET (MCRC, MM, ICUT)

IF (ICK.LT.0) GO TO 75

C  CHECK ON PLOT CF EDITED VALUES

IF (ICLT (NMC) .AND. IALL.E.G.C) EC TO 70

IF (ICLT (NCF).E.G.F.ITYP .AND. ITYP.E.G.0) GC TO 70

NS = 1

IF (ILEV.E.G.C) GC TO 44

DC 43 N = 1.MEY

IF (CLIT (LIST(E)).E.G.CCHR) GC TO 43

IADD = LCQS (N)

GO TO 56

43 CONTINUE

GO TO 70

44 CONTINUE

C  PICK ON SPECIFIC LEVEL AND CHARACTER

DO 45 N = 1.MEY

IF (LIST(E).E.G.LIT (*P 0.0)) GO TO 45
206 I=LCCS(N)
210 IF (ILEV.EQ.ICLT(I)) GO TO 31
211 IF (CLEV.EQ.ICLIT(ICLT(I))) GO TO 31
212 GO TO 45
213 31 CONTINUE
214 NS=1+1
215 GO TO 46
216 45 CONTINUE
217 GO TO 70
218 46 CONTINUE
219 DC 47 N=NS*#KEYS
220 IADD=LCCS(N)
221 IF (CLIT(LIST(N))).EQ.CCHR) GO TO 48
222 47 CONTINUE
223 GO TO 70
224 48 CONTINUE
225 C LOOK FOR THICKNESS
226 IAC2=0
227 IF (IL2.EQ.0) GO TO 56
228 DC 52 N=1*#KEYS
229 IF (.NOT.LIST(N)) RE.LIT('P') GO TO 52
230 I=LCCS(N)
231 IF (IL2.EQ.ICLIT(I)) GO TO 51
232 IF (CLEV.EQ.CLIT(ICLT(I))) GO TO 51
233 GO TO 52
234 51 CONTINUE
235 NS=NS+1
236 GO TO 53
237 52 CONTINUE
238 GO TO 70
239 53 CONTINUE
240 DC 54 N=NS*#KEYS
241 IADD=LCCS(N)
242 IF (CLIT(LIST(N))).EQ.CCHR) GO TO 55
243 54 CONTINUE
244 GO TO 70
245 55 IF (ICLT(IAC2).EQ.MISS) GO TO 70
246 56 IF (ICLT(IACD).EQ.MISS) GO TO 70
247 C LOCATE RASTER AND FIXEL OF SOUNDING
248 FLAT=ICLT(NLAT)*.0001
249 IF (FLAT.GT.LAN.OR.FLAT.LT.LAS) GO TO 70
250 FLON=ICUT(NLCN)*.0001
251 IF (FLON.GT.LCN.OR.FLON.LT.LCE) GO TO 70
252 IF (LUCF(1)).EQ.0) GO TO 12
253 C REVERT TO -L+E CONVENTION
254 SLOX=FLON
255 CALL SATEAR(FTIME,FLIN,FLEL,FLAT,SLON,2,INAV,PETAIN,SETDCT,0.)
256 IL=FLIN+.05
257 IE=FLEL+.05
258 GO TO 13
259 C USE TOVS NAVIGATION ROUTINE
260 12 LATS=FLAT 100.
LOADS=FLCN+1CC.
CALL SRCH(LATS,LONGS,IM,IL,IE,NFTS,NROWS)
IF (IN<EO)GO TO 70
C GUARD AGAINST OVERLAP IN SRCH TO AVOID DUPLICATION
IL=LT+LCF-1
IF (IL<LT+LCF*CR+IL+GE-LCT)GO TO 70
KOUT(1)=4
KCUT(2)=IL
KCUT(3)=IE
KCUT(4)=ILALC(FLAT)
KCUT(5)=ILALC(FLOI)
C IF (KFLG<NE<0)CALL CUTINT(KOUT)
CALL SATV1(IFFM+IL,IE,IRAS,IPIC+8,JC+CT)
C DC FLOATING
KACR=MCC(1CLT(NYF),10)+1
IF (KCLCR<CT<3)KCLCR=1
IF (KCLCR<LT<1)KCLCR=1
ICAT=ICUT(IADJ)/ISC
IF (IADJ<NE<0)ICAT=ICUT(IADJ)/ISC
KACR=1KWF(*CCLCR*1,KCLCR)
IF (LCR<NE<0)*IN*GO TO 60
SPD=FLCAT(ICAT)
IDIR=ICUT(IADJ-1)
DIR=FLOAT(IDIR)
YP=IRAS+MAG
XP=IPIC+MAG
IF (ISS<EC<0)GO TO 57
FLOI=-FLCN
ADJ=DIR
CALL CIRAC((FLAT,FLOI,SATLN,DIR,ADJ)
GO TO 58
SEC=1+CCS(FLAT+0174533)
ADD=APSLN+CCSI SEC)/0174533
IF (NOCE<EC<2)ADD=-ADD
DIR=DIR+ADD
IF (DIR<CT<360)DIR=DIR-360.
IF (DIR<LT<CT)DIR=DIR+360.
ADJ=DIR
SZ=MAG
IF (FLAT<LT<CT)SZ=MAG
CALL BAFB(AE,SPC,XP,YP,KCLCR,SZ)
GO TO 70
C IF (ISLB<EC<0)GO TO 65
IF (ILCT(IADJ+ISUB)*EO+MISS)GO TO 70
C FORM DIFFERENCE QANTITY
ICAT=ICAT-ICLCT(IADJ+ISUB)/ISC
GO TO 67
C IF (CCHR<FG<1*CT*ISF+CCHR<FG<1*CT*ICAT=ICAT-273
C ABOVE TO CONVERT TO CENTI
IF (CCHR<FG<1*2)IDAT=(ICAT+5)/10
IF (CLEV<EC<1*DIFF)IDAT=IDAT 10
CONTINUE
IF (CHR.NE.'FOC') GO TO 68
KCLOR=KOP/IDAT/10
IDAT=IDAT/10
68 CONTINUE
ICAT=MCD(IDAT.1000)
CALL PLODIC(IRAS+MAG,IPIC+MAG,IDAT,MAG,1,KCLOR)
70 MM=MM+1
IF (MM.LE.MM.MAX) GO TO 15
80 CALL ENDFLT
RETURN
800 CALL STEST('PARAM') (P-LEVEL MB).9,0)
RETURN
900 CALL STEST('CANNOT OPEN MCFILE NO. *',MENO)
RETURN
902 CALL STEST('CANNOT LOCATE ROW NO. *',MRD)
RETURN
904 CALL STEST('TRCUELE READING DATA RECORD NO *',M)
RETURN
END
COMMON /DOC/IDOC(112)
COMMON /NAVD/FLAT,FLCN,VZEN,SZEN,IL,IE,IFAS,IPIC,ITIME,JTIME,TCAY
COMMON /EBLG/KEUG
COMMON /DIMEN/NROWS,NCOLS
COMMON /THICK/LF2
CHARACTER * MFIL
CHARACTER 12 CPF,CLIT,CLEV,CKWP,ICH,LCHR,CGS
REAL * CLIT
REAL * FA,SA,SLM,SLMS
DIMENSION KCLT(6),MCHD(64),KBUF(400)
DIMENSION LEFL(33)
DIMENSION ITG(2400),IDG(2400)
DIMENSION ILN(4),IREP(3000),IA(3000),FA(3000),CL(3000)
DIMENSION CAS(3000),STLMA(3000),STLCN(3000)
DIMENSION FLN(2400),LT1(2400),WT2(2400)
DIMENSION IHR(400),ISCL(400),IUM(400),LCMS(400)
C----DESCRIPTION OF 4-WORD GRID HEADERS
C
INTEGER IHID(64)
INTEGER ICID(8)
C----- GIVE TOTAL SIZE (ICFRCS), # ROWS, # CCLS (IGSIZE=IGNP=IGNC)
C----- EQUIVALENCE (ICGSC,HIGC(1)),(IGNR,HIGC(2)),(IGNC,HIGC(3))
C----- YCOCD, HMMSS AND VALID-TIME (IF APPLICABLE) FOR GRCI
C----- EQUIVALENCE (IGCA,YIGC(4)),(IGTIME,YIGC(5)),(IGTMV,YIGC(6))
C----- DESCRIPTION OF GRIDDED VARIABLE (IN ME-FILE TERMS):
C----- NAME, SCALE, AND UNITS
C----- EQUIVALENCE (IGVMAP,YIGC(7)),(ICVSCA,YIGC(8)),(IGVUNI,YIGC(9))
C----- DESCRIPTION OF VERTICAL LEVEL: VALUE, SCALE, AND UNITS
C----- EQUIVALENCE (IGLEVr,YIGC(10)),(IGLSCA,YIGC(11)),(IGLNI,YIGC(12))
C----- GRIDDED-VARIABLE TYPE: =1 (TIME DIF) 2 (TIME AVG) 4 (LEVEL DIF)
C----- 0 (LEVEL AVG) OR ANY SUM OF THE FOREGOING
C----- EQUIVALENCE (IGVTF,YIGC(13))
C----- FOLLOWING LSEC IF PARAMETER IS A VERTICAL (LEVEL) DIF OR AVG
C----- (SAME SCALE AS ILEVEL)
C----- EQUIVALENCE (IGCLIF,YIGC(14))
C----- FOLLOWING LSEC IF PARAMETER IS A TIME DIF OR AVG (HMMSS)
C----- EQUIVALENCE (IGCTIF,YIGC(15))
C----- GRID ORIGIN, TYPE (I.E. TYPE OF PROJECTION)
C----- EQUIVALENCE (ICGRID,YIGC(33)),(IGGRID,YIGC(34))
C----- SUBSEGMENT CDS (IGLMX,IGLMPX,IGLAMX,IGLCMX,IGNCR)
C----- IGPC8R, IGFCLC, IGSP60, IGCLON) ALL HAVE 4 IMPLICIT DEC. PLACES.
C----- LAT GOES FROM -900000 TO 900000, LON GOES FROM -1800000
C----- TO 1800000 (WEST IS +)
C----- TYPE 1 GRIDS ARE PSCLD-MERCATOR
C----- EQUIVALENCE (IGCLAM,YIGC(35)),(IGLCMPX,YIGC(36)),(IGLMNP,YIGC(37))
C----- , (IGLCMPX,YIGC(38)), (IGNCPR,YIGC(39))
C----- TYPE 2 GRIDS ARE PLANAR-STEREOGRAPHIC
C----- GIVE ROW # CF NORTH POLE, CCL # OF N.P., CCL SFlACING AT 60 DEG N
C----- (DEC), LONGITUDE PARALLEL TO COLUMNS (DEC)
C----- EQUIVALENCE (IGFCFL,YIGC(35)), (IGFCULC,YIGC(36)), (IGSP60,YIGC(37))
C----- , (IGCLOA,YIGC(38))
C----- INITIALS OF USER AND PROJECT # UNDER WHICH GRID CREATED
C----- EQUIVALENCE (IGUSER,YIGC(41)), (IGPRG,YIGC(42))
C----- CHARACTER IC SUPPLIED BY PROGRAM (AREITPARY)
C----- EQUIVALENCE (IGIC,YIGC(43))
C----- EQUIVALENCE (IGT2(1),IT6(1))
C----- DATA MISS/2800/0080/NSIZE/2400/NSMAX/5000/
C----- DATA LP/R3-Z40404040/ILBNK/Z40404040/
C----- DATA FILE, LEN/100/VASTEXT *20,100/
C----- IALL=IKLP(*)ALL*,1,0)
C----- NGOUT=IKLP(*NGC*,1,0)
C----- 159 IS MAX NC. CF GRIDS PERMITTED BY IG SOFTWARE
C----- IF (NGCUT,NE,0)NGCUT=-NGCUT
C----- ABOVE FORCES OUTPUT GRIC NC
C----- EIAS=IKLP('EIAS',1,0)
C----- KFCG=IKLP('ELG',1,0)
C----- IF (KFCG,GE,0)CALL SCSEST (* GC!!!!!!!!!!!!,0)
C----- PICK UP CONTEXT INFORMATION
C----- ITERM=LUC(2)
C----- IF (LUC(81),NE,C)GC TO 111
105 CALL TRSIP(1,1,1,1,1,1,ITCC)
106 GC TO 112
107 111 CALL DCFCNM(DCIT(MPIF),LUN,LEN)
108 CALL DREAD(LLN,ITERK,ITCC)
109 CALL DCLSEE(LLN)
110 112 MDNC=1KWF(*MCNR*,1,0)
111 IF(MDNC*EC,0)MDNC=ITCC(40)
112 NGFC=1KWF(*NGFC*,1,0)
113 IF(NGFC*EC,0)NGFC=ITCC(42)
114 MDR=1KWF(*MCHR*,1,0)
115 IF(MDR*LT,0)MDR=ITCC(41)
116 MDNC=ITCC(38)
117 MDNG=1KWF(*MDNG*,1,MDNG)
118 ICHR=CFI(1,'Z')
119 LCHR=ICHr
120 ILEV=1PP(2,0)+10
121 LPR=P10
122 IF(MCNFGC(MDNC,MCHR,*NE,0)GO TO 500
123 IF(MDCPEN(MDNC,2,0)*NE,0)GO TO 500
124 C READ ROW HEADER RECORD
125 ICK=MDGFT(MDNC,MDF,M,KBUF)
126 IF(ICK*,LT,0)GO TO 502
127 ISYC=KBLF(1)
128 IHMS=KBUF(2)
129 C READ TEST RECORD
130 M=0
131 1 M=M+1
132 IF(M*,GT,100)GO TO 504
133 C READ COLUM HEDER FCR RAOE
134 C IF(MHDCE,0)ICK=MDGET(MDNC,M,N,KBUF)
135 ICK=MDGET(MDNC,MDF,M,KBUF)
136 IF(ICK*,LT,0)GO TO 1
137 MMAX=MDHC(E)
138 IF(KLOG*,GT,0)CALL SDEST(* MMAX IS * MMAX)
139 C MMAX IS MAX POSSIBLE DATA ENTRIES USED TO AVECIE
140 C INFINITE LCCPING AT SN 148
141 C FIND ADDRESS FOR LAT AND LON
142 NKEYS=MDKEYS(MDNC,-1,LIST,ISCL,LUN,LOCS)
143 NCA=0
144 DC 2 N=1,NKEYS
145 IF(LIST(N),EC,LIT(*MOD *))NMOD=N
146 IF(LIST(N),EC,LIT(*CMAX *))NCA=N
147 IF(LIST(N),EC,LIT(*NREC *))NCA=N
148 IF(LIST(N),EC,LIT(*LAT *))NLAT=N
149 IF(LIST(N),EC,LIT(*LOM *))NLON=N
150 C SAVE REPRESENTATIVE ENTRIES FOR ACDATA CASE
151 IF(LIST(N),EC,LIT(ICHRE))NACC=N
152 C CONTINUE
153 NRPT=500
154 IF(NCA*,NE,0)NRPT=KBUF(NCA)
155 NRPT=1KWF(*LAST*,1,1,NRPT)
156 IF(LCHR,EC,'U',OR,LCHR,EC,'V')ICHRE='SFD'
157     IF IAS=0
158     IF (ICHF*EG.*(T*)*.CP.*ICHF*EG.*(TC*'))
159         IAS=27312
160     C CHECK ON KEYIN CVERRIDE
161     IF (FIAS.NE.0)IEIAS=EIAS
162     ISUP=0
163     IF (ICHF.NE.*TCIF*)GC TO 20
164     ISUP=5
165     ICHR=(*T*')
166     20 IF (ICHF.NE.*CCIF*)GO TO 30
167     ISSUE=F
168     ICHR=(*TD*')
169     30 CONTINUE
170     *S=1
171     C CHECK FOR SELECTIVE PRESSURE
172     CLEV=CPP*(2,*')
173     IF (ICHF*EG.*(Z')*ANC*ILEV.EG.0*ANC*CLEV.EG.*(')')ILEV=5000
174     LPR=ILEV/10
175     MMCC=ILEV
176     IF (ILEV.EG.0)MMCC=599
177     C CHECK FOR THICKNESS CALCULATION
178     IL2=IKWP*(ILEV.1.0)*10
179     LPR2=IL2/10
180     C CHECK FOR TEB DIFF
181     IF (CLEV.NE.*DIFF')GC TO 10
182     ISUB=13
183     C CHECK FOR GRESS GRIC
184     10 NGG=1PP*(3,0)
185     IF (NGG.NE.0)GO TO 14
186     C CHECK OPERATOR FORCED BOUNDARIES
187     LLNW=IDCC(25)
188     LLSE=IDCC(26)
189     LN=LLNW/1000
190     LS=LLSE/1000
191     LW=MOD(LLNW.1000)
192     LE=MOD(LLSE.1000)
193     L=IKWP(*LAT.1.0)
194     IF (L.NE.0)LS=L
195     L=IKWP(*LAT.2.0)
196     IF (L.NE.0)LM=L
197     L=IKWP(*LCN.1.0)
198     IF (L.NE.0)LE=L
199     L=IKWP(*LCN.2.0)
200     IF (L.NE.0)LW=L
201     LW=IABS(LW)
202     LE=IABS(LE)
203     IF (LW.GT.160)LW=LW-360
204     IF (LE.GT.160)LE=LE-360
205     IF (LW.LT.LE)LW=LE+360
206     LINC=IKWP(*INC.1.0)
207     DINC=1 FLCAT(LINC)
208     RCWS=LA-LS
CCLS=LW-LE
FTS=(ROWS+1)* (CCLS+1)
C Establish grid increment in 10ths of degrees
FN=FLCAT(NSIZE)
IF (LINC.EQ.0) GO TO 14
INC=INC+1C+6
LAN=LN 10
LAS=LS 10
LCW=LW 10
LOE=LE 10
DINC=INAC
DLAT=DINC
DLCN=DINC
NROWS=(LAN-LAS)/INAC+1
NCCLS=(LCW-LOE)/INAC+1
NPTS=NROWS*NCCLS
IF (LINC.NE.0) GO TO 14
IF (NPTS.LE.NSIZE) GO TO 14
INAC=INAC+1
GO TO 13
14 CONTINUE
C USE IDATA AS FLAG IN EVENT Cf AC DATA
IDATA=77777
C INITIALIZE ANALYSIS FIELD
DC 12 L=1,NSIZE
IDL(L)=C
ITGCL=0
IGE=0
IF (NGC.EQ.0) GO TO 105
ICK=IGEET(NGFC,NCG,2400,IDL(1),NROWS,NCCLS,IGFC)
IF (LIT(ICHF).NE.IGVNAM) GO TO 940
C ABOVE COMMENTED OUT UNTIL CALLAN JOINS THE REST OF LS
CALL SDESTE(LPR IS ',LPR)
CALL SCESTM(IGLEV IS ',IGLEV)
C IF (LPR.NE.IGLEV) GO TO 940
C NOTE. ABOVE WILL DEFINE THE GRID SIZE
IF (IOK.LT.0) GO TO 300
NPTS=NROWS*NCCLS
SCL=10. *IGWSCA
DC 102 N=1,NPTS
102 FCN(K)=FLCAT(ILN(K))/SCL
LAN=IGLAPY/1000
LAS=IGLAMA/1000
LCW=IGLCXX/1000
LOE=IGLCXN/1000
INAC=IGINCRC/1000
DLAT=INAC
DLCN=INAC
IGE=1
105 TLAT=0.1:FLCAT(LAN)
SLAT=0.1:FLCAT(LAS)
WCLN=0.1:FLCAT(LIOW)
ELOC=C:1:FLC(T(LCE)
DINC=DINC:G:1
DLAT=DLAT:G:1
DLOC=DLOC:G:1
IF(KBUCE.GE.0)CALL SDEST(AROWS,NCOLS,INC:0)
KOUT(1)=3
KOUT(2)=AROWS
KOUT(3)=NCOLS
KOUT(4)=INC
IF(KBUCE.GE.0)CALL CUTINT(KOUT)
C
CHECK GUESS QFTIGA..FILL FRCM *VASSS*
CGS=CGS+1,1,G
IF(CGS.NE.(*G*))GO TO 110
IGE=1
NCSEC=NCSEC+1,0
CALL ANGSS(IDL,ITG,NCOLS,AROWS,TLAT,WLCM,DINC,LFR,
     ICHR,KMENG,NCSEC)
110 NE=0
C
NB IS USEC TC COUNT REPORTS FOR BARNES ANALYSIS
IF (IKUF(NCCAT:1,0).NE.0)GO TO 150
IF (KBLG.GT.0) CALL SDEST(* NO. CF REPORTS IN FILE IS *NRPT)
SUM=0.
SUM'=0.
C
BEGIN IMPLICIT GO LOOP ON REPORTS
MN=1
C
NN IS USEC TC COUNT ALL REPORTS
LL=0
C
LL IS USEC TC COUNT ALL ACCEPTED REF'S (INCLUDING CROSS ERRCR EDI
C
FCP THE FINAL EDIT STEP IF REQUESTED
KK=C
C
KK IS USEC TC COUNT ALL VALID REPORTS (WITH DATA)
C
(IN RACB FILES MANY "COLUMNS" DO NOT IN FACT REPORT)
IADD=0.
ERR=999999.
IF (ICHG.EQ.*TBE*) ERR=15.
IF (ICHG.EQ.*Z*) ERR=30.
IF (ICHG.EQ.*T*) ERR=6.
IF (ICHG.EQ.*TC*) ERR=3.
ERR=DKLF('ERR',1,ERR)
FCK=ERR-Z.
C
FCK IS GROSS EPRCR CHECK AGAINST GUESS FILE
15 CONTINUE
IF (MCFLC.EQ.0) GC TO 115
C
READ COLLMN HEADER FOR RACB
ICK=MCGET(MCAC,CAK,KELF)
IF (ICK.LT.0) GC TO 148
115 CONTINUE
ICK=MCGET(MCAC,MDF,AN,KELF)
IF (ICK.LT.0) GC TO 148
IF (KELF(NCF).EQ.0.AND.IALL.EQ.0) GC TO 148
C
SKIF REPORTS WHICH HAVE BEEN PREVIOUSLY REJECTED
NS=1
IF (ILEV.EQ.0) GO TO 44
C Continue
GO 43 N=1,AKEYS
IF (LIST(N).EQ.LIT(ICH)) GO TO 45
KS1=N
GO TO 48
CONTINUE

C Pick on specific level and character
DC 46 N=1,AKEYS
IF (LIST(N).EQ.LIT(*)&GO TO 45
I=LCCS(N)
IF (ILEV.EQ.KELF(I)) GO TO 31
IF (CLEV.EQ.CLIT(KELF(I))) GO TO 31
GO TO 46
CONTINUE

KS=N+1
GO TO 46
CONTINUE
GO TO 148
CONTINUE
GO 47 N=NS,AKEYS
KS1=N
IF (LIT(ICH).EQ.LIST(N)) GO TO 48
CONTINUE
GO TO 148
CONTINUE
K7=KK+1
C Look for thickness
IAD2=0
IF (IL2.EQ.0) GO TO 56
GO 52 N=1,AKEYS
IF (LIST(N).EQ.LIT(*)& GO TO 52
I=LCCS(N)
IF (IL2.EQ.KELF(I)) GO TO 51
IF (CLEV.EQ.CLIT(KELF(I))) GO TO 51
GO TO 52
CONTINUE
N=R+1
GO TO 53
CONTINUE
GO TO 148
CONTINUE
GO 54 N=NS,AKEYS
IAD2=LCCS(N)
IF (LIT(ICH).EQ.LIST(N)) GO TO 55
CONTINUE
GO TO 148
IF (KELF(IAD2).EQ.MISS) GO TO 148
CONTINUE
IADD=LCCS(NS1)
KCUT(1)=4
KCU T ( 2 ) = K L F ( A L A T )
KCU T ( 3 ) = K L F ( A L C N )
KCU T ( 4 ) = K L F ( I A C D )
KCU T ( 5 ) = K L F ( I A C C )

IF ( K E U G C T * 1 ) CALL CU TI N ( K C U T )
IF ( K E U F ( I A C D ) * E G * M I S S ) G O T O 1 4 6
I S = I S C L ( N 1 )
F S C L = 1 0 . + I S
I D A T = K E U F ( I A C C )

IF ( I A D 2 . * # E ) I CAT = I D A T = K B U F ( I A D 2 )
F L A T = K E U F ( A L A T ) * 0 0 0 1
F L C N = K B U F ( A L C N ) * 0 0 0 1

IF ( F L A T * C T * T L A T * G R * F L A T * L T * S L A T ) G O T O 1 4 8
F L A T J ( T L A T - F L A T ) / C L A T + 1 . 0
I F ( F L O N * L T * E L O N * G R * F L O N * G T * W L O N ) G O T O 1 4 8
F L O N I ( W L C N - F L C N ) / C L C N + 1 . 0
I D A T = I C A T * I E I A S
I F ( I S L B . * N # 0 ) I C A T = I C A T = K B U F ( I A C C + I S L B )
D A T = F L C A T ( I C A T ) / F S C L
L L = L L + 1
D A ( L L ) = D A T
S T L A T ( L L ) = F L A T
S T L C N ( L L ) = F L C N
I R E P ( L L ) = I N

C A B O V E A R E S A V E D F O R E D I T I N G
C A B O V E I S G R O S S E R R O R C H E C K

IF ( ( C C C . # E L G . / F L ) ) G O T O 1 4 6
I F ( L K W P ( * E D I T . * 1 0 ) . * L T . L ) G O T O 1 4 6
I F ( A B S ( V A L - C A T ) . * L T . F C K ) G O T O 1 4 6
C A B O V E I S G R O S S E R R O R C H E C K

K C U T ( 1 ) = 4
K C U T ( 2 ) = F L A T
K C U T ( 3 ) = F L C N
K C U T ( 4 ) = D A T
K C U T ( 5 ) = V A L
C A L L C U TI N ( K C U T )
G O T O 1 4 8

1 4 6 I F ( I C H R * E G * # E I R * A N D . C A T . L T . 0 ) D A T = C A T + 3 6 0 .
I F ( I C H R * E G * # S F C * ) G O T O 7 0
I F ( L C H R * E G * # S F C * ) G O T O 7 0
S P D = D A T
I D I P = K B L F ( I A C D - 1 )
L S = I S C L ( N 5 1 - 1 )
S S C L = 1 0 . - L S
D I F = F L C A T ( I I C I R ) / S S C L * 0 1 7 4 5 6 3
U = S P D * S I N ( C I R )
V = S P D * C O S ( C I R )
D A T = U
I F ( L C H R * E G * # V * ) C A T = V
C O N T I N U E
ADD ARRAY FOR BARRIES
FILL BARRIES ARRAYS
\#\#NB+1
DAS(NP)=DAT
THE VALUES OF DAS WILL BE CHANGED IN FBARN
RW(NB)=FLATW
CL(NB)=FLCN1
COMPUTE STATISTICS
SUM=SUM+DAT
SUMS=SUMS+DAT+DAT
IDATA=0
IF (NP.LT.NMAX) GC TO 140
CALL SDEST(* TOC MUCH DATA...EXITING TO ANALYSIS,0)
GC TO 140
M=NN+1
IF (KK.GT.KRFT) GC TO 140
IF (NN.LE.MM1X) GC TO 15
IF (FRUG.GT.3) CALL SDEST(* ACTUAL NO. OF RFTS IS N,NN)
IF (IDATA.EQ.77777.AND.IGB.NE.1) GC TO 400
IF (IDATA.EQ.77777) GC TO 150
IF (FN.LT.2) GC TO 145
FN=NB
SUM=SUM/FN
SUMS=SUMS/FN
SC=DSQRT(SLWS-SUM-SUM)
SD=AMAX1(ERR,SC)
SD=AMA11(I,SC)
ISO=SD 100
IF (ERR.EQ.5555555.) ERR=SD
IF (KELG.GT.0) CALL SDEST(*DAPA TOSSCUT 100 IS ISO)
SET BARRIES SCALING
ISC=IKW1(*SCL*1,SC)
IF (COS.NE.(*G*)) GC TO 145
DC 135 N=1,KRTS
FLC(N)=FLCAT(IDL(N))/FSCL
CALL FEAPN(TLAT,NRCWS,ACCLS,FLD,WT1,WT2,DAS,RL,CL,
NE,DLAT,ISC,IGB)
DC 155 N=1,KRTS
IDL(N)=FLC(N):FSCL*0.5
II=FSCL*100
KOUT(1)=5
KOUT(2)=IDL(10)
KOUT(3)=IDL(20)
KOUT(4)=IDL(30)
KOUT(5)=IDL(40)
KOUT(6)=IDL(50)
IF (KELG.GT.1) CALL CUTINT(KOUT)
IF (KELG.GT.1) CALL SDEST(* SCALE 100 IS II)
CHECK ON THE EDIT CFTIA
IF (IK*P.ECITY*1,0).LE.0) GC TO 150
NOTE...WE EDIT ALL REPORTS THAT ARE CURRENTLY VALID EVEN IF
THEY DO NOT TAKE PART IN THE ANALYSIS
DO 130 NN=1,LL
FLAT=STLAT(AN)
FLC=STLC(AN)
CGET VALUE FROM GRID
CALL VALUE(LAN,LAS,LCW,LEC,INC,ICL,FSCL,VAL)
**VAL=**VAL
**DAT =**DAT(AN)
**DIF =**DIF-VAL
C CROSS ERROR CHECK
ADIF=A2S(DIF)
IF (ADIF.LT.ERR) GO TO 130
**ICK =MDGET(GNCF,MCF,IREF(AN),KEUF)
**KEUF(KMCC)=**MCC
**ICK =MDFLT(GNCF,MCF,IREF(AN),KEUF)
IF (KBLG.LT.C)GO TO 130
**D =**DAT-VAL
CALL ERKCEE(1Y,IE,2F7.2,3F10.1/)',LBF,NN,FLAT,FLC,DAT,VAL,C)
130 CONTINUE
RETURN
C OUTPUT FILEC
150 CONTINUE
IF (IACC.EQ.C)IADD=ADD
IGVMAP=LIT(LCPR)
IGVSCA=ISCL(IADD)
IGVUNI=ILN(IADD)
IGLEVEL=LPR
IGLS=1
IGLUNI=1
IGVTYP=F
IGLIF=1
IGCRG=0
IGTYPE=1
IGLAMX=LAN 1000
IGLAMX=LAS 1000
IGLCM=LOW 1000
IGLCM=LOE 1000
IGLINC=INC 1000
IGDAY=MC(G(IYEC,10000))
ICTIME=IF**100
IGSIZE=NPTS
ICF=NRWS
IGNL=NCLS
NGRF=LLC(6)
C FOLLOWING IS TO STORE RESULT IN SAME GRID FILE AS GUESS GRID
C AS EXPECTED BY "UCWA"
IF (NGC.NE.0)ANGF=NGFG
**ICK =IGPUT(NGF,NCLS,ILD,NPCLS,NCLS,IGH,ISTAT)
**IF (KEUG.GE.C)CALL SSFEST(* ANALYSIS FILED AS GRID NO.*ISTAT)
**IGVUNI=IUBLK
**IGVSCA=2
**IF (IWP(*LT *,1,0).EC.0)GO TO 296
IGVAM=LIT('CT1')
DC 295 N=1,FST
IDL(N)=WT1(N)*10000.
295 CONTINUE
IF (NGCLT.LE.0) NGCLT=NGCLT-1
ICK=ICPUT(NGRF,NGCLT,IDL,NRWS,ACCLS,IGHC,ISTAT)
IF (K&LGE.0) CALL SDEST(' WEIGHT1 FILED AS GRID NO.*,ISTAT)
296 IF (ICK.EQ.('CT ',2,'O',EC,'O')) RETURN
IGVAM=LIT('CT2')
DC 298 N=1,FST
IDL(N)=WT2(N)*10000.
298 CONTINUE
IF (NGCLT.LE.0) NGCLT=NGCLT-1
ICK=ICPUT(NGRF,NGCLT,IDL,NRWS,ACOLS,IGHC,ISTAT)
IF (K&LGE.0) CALL SDEST(' WEIGHT2 FILED AS GRID NO.*,ISTAT)
RETURN
300 CALL SDEST(' CANNOT READ GUESS GRID NO.*,NCG)
RETURN
305 CALL SDEST(' NC DATA AVAILABLE FOR IMAGE *,C)
RETURN
310 CALL SDEST(' CANNOT OPEN MDFILE NO.*,MDNC)
RETURN
315 CALL SDEST(' CANNOT LOCATE ROW NO.*,MRD)
RETURN
320 CALL SDEST(' TRICUBED READING DATA RECORD NO.*,L)
RETURN
325 END
1 //GWVAE660 USE CLASS=6, MSGLEVEL=(C,G)
2 // VLGWVA 90 02/28/64: MEMBER UPDATED
3 // VLGWVA SPG 12/23/63; ENTERED USER MANUAL CARE
4 // EXEC MCPKG, MCE-GLWA
5 //FORX, SYSIN DD
6 //EROSSE SS(TEKES, EMES, DMES, SPAD, ENCDE, DECODE, LWGET, LWPUT, LWCLS)
7 //EROSSE SS(ISFILE, LTRC, OFCCOM, SCX, SGW)
8 //EROSSE SS(MOVE, MCVC, MCVW, CLEANW)
9 //EROSSE SS(DOCEN, CREAD, CRWITE, ISFILE, LTRC, OFCCOM, SCX, SGW)
10 //EROSSE SS(MCVFEC, MCVGET, MCVPUT, ISCHAR, ISAN, MCVC, MCVC, MCVW, CLEANW)
11 //SUBROUTINE MAIN
12 C ** PROGRAM TO PROCEDURE WINDS FOR MD USING GRID HEIGHTS
13 C ** MD LINKAGE THRU VASTEXT FILE
14 C ** GWVA NGlev <KEYLYFDCS>
15 C ** NG - GRID NC (GRID FILE LINKAGE IS THRU "IGU SET AN")
16 C ** LEV- LEVEL(MP)
17 C ** IF LEVEL NOT SPECIFIED WINDS ARE NOT WRITTEN TO OUTPUT
18 C ** KEYWORDS:
19 C ** COLOR = COLOR OF WIND BARB PLOTTED ON IMAGE (1,2,3)
20 C ** DELT = MP OF HOURS BETWEEN NG (LATER TIME) AND NGCLD
21 C ** NGCLD = GRID NC OF PREVIOUS ANALYSIS
22 C ** TYPE = CONTROLS WIND DERIVATION (GGRD DEFAULT), AG, IS)
23 C ** SFAC = FACTOR TO MULTIPLY WIND SPEED (DEFAULT=1.0)
24 C ** PLT = NCR ZERO SKIPS PLOT
25 C ** SEC/NCAS USERS MANUAL - CHAP12
26 DIMENSION KCUT(10), MCVC(64)
27 DIMENSION WFILE(2400), WFILE(2400), WFILE(2400), ITAB(16)
28 DIMENSION ILEF, ECO, IREPRESS(10), SATPOS(10), INDEX(10)
29 DIMENSION LIST(10), ISCL, ISCL, ILM, ILC, ILC
30 DIMENSION CHARACTER * FILE
31 CHARACTER*12 CWTP, CTYP
32 COMMON /NAV/FLAT, FLON, ZENLOC, ZEN, IL, IE, IRAS, IFIC, IFMS, JT, JC
33 COMMON /LTYPE/KCEC
34 COMMON /SAT/C, C, SLAT, DINC
35 COMMON /SIZE/APT, MCFF, DXFC
36 COMMON /IMGCAR, AC
37 COMMON /COC/OCOC(112)
38 COMMON /TERMA/ITERP
39 COMMON /ANAL/NCAN, LTOP
40 COMMON /CREIT/ACCE
41 COMMON /TIET/ITCL
42 COMMON /ENTRY/INIT
43 EQUIVALENCE (IDCC(40), MDCR), (IDCC(41), MDRR)
44 INTEGER ICHC(64)
45 INTEGER IIGC(6)
46 C------GIVE TOTAL SIZE (WAREA), # ROWS, # CCLS. (ICSIZE=IGAR, IGC)
47 EQUIVALENCE (IGC=ICH(1)), (IGAR, IGH(2)), (IGNC, IGH(3))
48 C------YVDDCD, PMPSS AND VALID-TIME (IF APPLICABLE) FOR GRID
49 EQUIVALENCE (IGCAY, IGH(4)), (IGTIME, IGH(5)), (IGTIMV, IGH(6))
50 C------DESCRIPTION OF GRIDDED VARIABLE (IN MC-FILE TERMS):
51 C------ NAME, SCALE, AND UNITS
52 EQUIVALENCE (IGVMAP, IGH(7)), (IGVSCA, IGH(8)), (IGVUMI, IGH(9))
C-----DESCRIPTION OF VERTICAL LEVEL: VALUE, SCALE, AND UNITS
54 EQUIVALENCE (IGLEV,IGHD(14)),(IGSLCA,IGHD(15)),(IGLUNI,IGHD(12))
55 C-----GPIGED-VAEEABLE TYPE: =1 (TIME CDF) 2 (TIME AVG) 4 (LEVEL CDF)
56 C----- 8 (LEVEL AVG) OR ANY SUM OF THE FOREGOING
57 EQUIVALENCE (IGVTYPE,IGHD(3))
58 C-----GRID ORIGIN, TYPE (i.e., TYPE OF PROJECTION)
59 EQUIVALENCE (IGCCG,IGHD(33)),(IGCTYPE,IGHD(34))
60 EQUIVALENCE (IGLAMY,IGHD(35)),(IGLOFMX,IGHD(36)),(IGLAMN,IGHD(37))
61 ,(IGLEMN,IGHD(38)),(IGICR,IGHD(39))
62 DATA SATPCES/135.,75./
63 DATA DSIZE/E ./
64 DATA IFPRES/100.,50.,100.,250.,300.,400.,300.,250.,200.,150.,100./
65 C ABOVBE EQUATES PRESS TO RETRIEVAL LEVELS
66 DATA ILEN/18/.
67 C ABOVBE IS THE NO. OF WORDS IN THE ROW HEADER
68 DATA MFILE/*VASTEXT */LUN/20/LEN/100/
69 DATA SFAC/1./,IFP/O/
70 C
71 CALL INITPL(C,C)
72 C FORCE HIGH RESOLUTION SRC
73 ITOL=1
74 JR=LUC(-1)
75 IPLT=IKWF("PLT",1,0)
76 IF (LUC(81).EQ.0) GC TO 120
77 C READ VASTEXT CONTEXT FILE
78ITERM=LUC(-20)
79 CALL DCOPEN(MFILE,LUN,LEN)
80 CALL DREAD(LUN,ITERM,IDCC)
81 IF (IPLT.NE.0) GC TO 130
82 C INITIALIZE NAVIGATION
83 IRAS=LUC(-11)
84 IFIC=LUC(-12)
85 CALL TVSAT(JFR,IRAS,IPIC,IL,IE,ISS,JC,IT)
86 JC=ISS ICICCCC+JC
87 CALL NVINIT(EETAIN,EETDCT,INAV,PTIME)
88 GC TO 130
89 120 CALL TSNIC(1,1,1,1,1,1,1,1,1,IDCC)
90 NODE=IDCC(7)
91 INIT=1
92 IF (IPLT.NE.0) GC TO 130
93 C SET UP AREA FOR IMAGE SPACE TO PLCT
94 CALL TVSAT(JFR,405,335,LTCP.LELE,ISS,IC,IT)
95 CALL TVSAT(JFR,495,335,LEOT.LELE,ISS,IC,IT)
96 IF (LTCP.LT.1) LTCP=1
97 IF (LBOC.CT.NRCPW) LBOC=NRCPW
98 MROWS=LBOC-LTCP+1
99 MPT=MROWS 56
100 130 CONTINUE
101 MENC=IDCC(40)
102 MDR=IDCC(41)
103 2 IF (MDCPEN(MENC,2).NE.0) GO TO 900
104 C CCGERIALIZATION
CTYP=CKUP(*TYPE*,1,*CR*)

ABOVE DEFAULTS TO ECMSTROPIC*GRADIENT ADJUSTMENT

SET TYPE CHICES THROGLH KGEC

KGEC=11
IF (CTYP=EG,*G*) KGEC=0001
IF (CTYP=EG,*AC*) KGEC=1111
IF (CTYP=EG,*IS*) KGEC=1100
IF (CTYP=EG,*CU*) KGEC=C100
IF (CTYP=EG,*MG*) KGEC=1000

ISAT=0
IF (LUC(81)=EG.0) GC TO P
NSAT=ISATNV(IACC(1))
SATLON=-SATFCS(*SAT)

KOL=IKWP(*CCOR*.1,1)
NGRF=LUC(6)
NGR=IPF(1,0)
IGK=IGGET(NGRF,ACR,2400,WHITE,NA,AC,IGHD)
IF (IGK.LT.0) GC TO 995
LEVEL=IGLEVEL
NLV=10

LEV=IPF(2,0)

C ABOVE WILL ALLW ALL WINDS TO PACT WHEN NOT STORING
IF (LEV.EG.0) GC TO 15
IF (LEVEL.EG.LEVF) GC TO 12
CALL SDEST(* GRID LEVEL DOES NOT AGREE WITH KEYEC IN** LEVF)
RETURN

12 CONTINUE
DO 14 I=1,10
IF (LEV.EG.IPRESS(I)) ALV=I

14 CONTINUE
15 CONTINUE

LAD=IGLAMX/1000
LCK=IGLCX/1000
LAS=(IGLAMX-IINC,AR)/1000
LCE=(IGLCX-IINC,N)/1000
INC=IINC/1000

C INC IS SPICELNGTH IN TENTHS OF DEGREES LAT
C THIS PROGRAM ASSUMES THAT INCREMENTS ARE NEVER FINER
INC2=INC/2

C GFT HORIZONTAL INCREMENT IN METERS
DINC=FLCAT(INC) 11100.

DX=DINC
DY=DINC

150 LAMAX= (LAM-2*INC-INCC2)/10
LONMAX= (LON-2*INC-INCC2)/10
LATMIN= (LAS+2*INC+INCC2) 10
LCMGIN= (LCE+2*INC+INCC2) 10
NPTS=NR*NC

DO 160 J=1,NPTS
J=IT2(J)=WHITE(J)
160 CONTINUE
157 IF ID=0
158 KWIN=0
159 C READ IN KEYS
160 IF (MCINC(MDNC,MCMD).NE.0) GO TO 900
161 NKEYS=MCKEYS(MDNC-1,LIST,ISCL,ILNK,LCCS)
162 C READ TEST RECORD
163 N=0
164 101 N=N+1
165 IF (M.GT.100) GO TO 904
166 C READ RCW HEADER
167 ICK=MCGET(MDNC,MCR,G,IBUF)
168 C READ COLUMN HEADER IF IT EXISTS
169 IF (MDCS(G).NE.0) ICK=MCGET(MDN0,G,M,IBUF)
170 ICK=MCGET(MDNC,MCR,M,IBUF)
171 IF (IOK.LT.0) GO TO 101
172 C FIND LAT/LON ADDRESS IN KEYLIST
173 NCA=0
174 DC 11 N=1,AKEYS
175 IF (LIST(N).EQ.LIT("CMAX"))NCA=N
176 IF (LIST(N).EQ.LIT(".REC"))NCA=N
177 IF (LIST(N).EQ.LIT("LAT"))NLAT=N
178 IF (LIST(N).EQ.LIT("LON"))NLCN=N
179 10 CONTINUE
180 LAST=500
181 IF (NCA.NE.0)LAST=IEUF(NCA)
182 LAST=IKWF(*LAST+1,LAST)
183 KBUG=IKWF(*ELG+1,0)
184 KOUT(1)=3
185 KOUT(2)=NCA
186 KOUT(3)=NLAT
187 KOUT(4)=NLCN
188 IF (KBUG.NE.0) CALL CUTINT(KOUT)
189 IF (KBUG.NE.0) CALL SDEST(* NC OF REPORTS IS *LAST)
190 IF (LAST.NE.0) GC TO 10
191 CALL SDEST(* NO DATA AVAILABLE.*0)
192 RETURN
193 10 CELT=DKWF(*CELT+1,G)
194 NGR=IKWF(*NGCR,G,1,0)
195 IF (CELT.NE.0) ICK=ICGET(NGRF,NGR,2400,WHIT2,NR,NC,IGCD)
196 IF (ICK.LT.0) GC TO 995
197 SFAC=DKWF(*SFAC+1,1,0)
198 IPR=0
199 C CONVERT HITES TO WHOLE METERS
200 ISC=10:ICNSCA
201 DC 165 J=1,NFTS
202 WHIT(J)=(WHIT(J)+5)/ISC
203 WHIT2(J)=(WHIT2(J)+5)/ISC
204 WHIT3(J)=(WHIT(J)+WHIT2(J))/2
215 C ABOVE IS FOR COMPLETING TRAJECTORY
216 CONTINUE
207 JC=IDOC(1)
208 JT=IDOC(2)
IFIND=0
MST=0
DC 200 N=1,LAST
IFIND=0
IF (MODF(E,9,.1,.0)ICX=MDCET(MCNO,C,N,IEUF))
ICX=MDCET(MCAG,MCR,N,IBLF)
IF (ICX.LT.0) GO TO 200
DC 110 L=1,NKEYS
IF (LIST(L).NE.LIT(*P,+)) GO TO 110
IF (IFINC.NE.0) GO TO 13
C SAVE SFC PRESSURE
NSFP=L
IFIND=1
10 I=LOC5(L)
C CHECK ON SPECIFIC LEVEL
IF (I=LEF(I)/10.E+LEVEL) GO TO 110
NWNL=I+3
110 CONTINUE
IF (NWNL.EQ.0.AND.LEVP.NE.0) GO TO 200
C AVOID WINCS EELCW GROUND
IF (IEUF(NSFP).LT.IPRESS(NLV)) GO TO 200
LAT=IBUF(LAT)/100
JLAT=LAT
IF (LAT.GT.LATMAX.CR.LAT.LE.LATMIN) GO TO 200
LCN=IBUF(ALCN)/100
IF (LOC.GE.LCNMCR.CR.LCN.LT.LCNMIN) GO TO 200
FLAT=IBUF(ALAT)+0001
SLAT=FLAT
IF (IPLT.NE.0) GO TO 168
FLON=IBUF(ALCN)+0001
IF (LUC(B1).EQ.0) GO TO 32
C REVERT TO -W+E CONVENTION
SLON=-FLON
CALL SAETAC(FTIME,FLIN,FELE,FLAT,SLON,2,INAV,BEATIN,BEDOT,0.)
IL=FLIN+0.5
IE=FELE+0.5
GO TO 33
C USE TOVS NAVIGATION ROUTINE
32 LATS=FLAT 1CC
LONCS=FLCN 1CC
CALL SRCF(LATS,LONCS,IM,IL,IE,NPT,ARCWS)
IF (IM.EQ.0) GO TO 200
C GUARD AGAINST OVERLAP IN SRCF TC AVOID DUPLICATION
IL=IL+LTCP-1
IF (IL.LT.LTCC.OP.IL.GE.LBCT) GO TO 200
33 CALL SATTV(FR,IL,IE,IF,J,F,S,J,CML,LT)
168 I=(LOW-LCN/10+INCC)/INC+1
J=(LAN-LAT/10+INCC)/INC+1
CALL ZWINC(WITE,I,J,JLAT,SPEED,DIREC,WHIT2,DELT,WHIT3)
ISPD=SPEED
IF (DIREC.LT.0) DIREC=DIREC+360.
SPEED=SPEED SFAC
IF (IPLT .NE. 0) GC TO 170
XP = IF
YP = JP
FLON = -.01 FLCAT(LCN)
ADJ = DIREC
IF (SAT .NE. 0) CALL DIREC(U,FLAT,FLCN,SATLKN,ADJ)
SZ = SIZE
IF (FLAT .LT. 0.) SZ = - SIZE
CALL BARB(ADJ,SPRED,XP,YP,KCL,SZ)
C CHECK FIGHT LEVEL FOR WRITING WIND CB
170 IF (LEVF .NE. 0) GC TO 175
IF (IPR .NE. 0) GC TO 200
IPR = 1
CALL SDEST(* LINES ACT WRITTEN TO OUTPUT FILE...*,0)
GO TO 200
175 CONTINUE
IDIR = DIREC + 0.5
ISPC = SPRED + 0.5
IBUF(NWIN) = ICIR
IBUF(NWIN+1) = ISPC
ICK = MDPUT(MNC,MCR,N,IBUF)
IF (IOK .LT. 0) GO TO 906
200 CONTINUE
300 CONTINUE
CALL ENDFLT
CALL SDEST(* DUN...*,0)
C RETURN
C 500 CALL SDEST(* CANNOT OPEN MCFILE NO. *,MDNO)
RETURN
904 CALL SDEST(* TROUBLE READING DATA RECORDS *,0)
RETURN
906 CALL SDEST(* TROUBLE WRITING DATA RECORD NO. *,N)
RETURN
995 CALL SDEST(* UNABLE TO READ GRID ICK= *,ICK)
RETURN
C END
SUBROUTINE MAIN

C? UPDATE THE CURRENT GUESS GRID FILE USING RETRIEVALS
C? GRID FILE PINTER DEFAULTS TO VASTEXT
C? KEYIN: LGCA <AGRID>
C? POSITIONAL PARAMETERS:
C? <AGRID> BEGINNING NO. OF GRIDS TO BE UPDATED (NO DEFAULT)
C? GRID FILE DEFAULTS TO GUESS GRID OF VASTEXT
C
SSEC/NCICAS USER MANUAL - CHAP12
COMMON /ICC/ICCCO(100)
DIMENSION LIST(26),KCUT(10),ISCALE(26),IUNIT(26),LOC(26)
C ABOVE TO RECEIVE MOSKEY INFIC
INTEGER A CELF(20)
DIMENSION IG(320),MES(64),JREC(26),IREC(26),FISC(22)
DIMENSION ICHAR(22),KCHAR(22),JCHAR(22),IFRESS(22),ICCT(22)
DIMENSION LCHAR(22)
INTEGER IGK(64),IGDET(64),IGID(8),ROW(3),COL
C-----DESCRIPTION OF ICE-LCD GRID HEADERS
C
C ----GIVE TOTAL SIZE (WCRDS), # RCLS, # CCLS. (IGSIZE=IGNR=IGNC)
EQUIVALENCE (IGSIZE,IGHC(1)),(IGNR,IGHC(2)),(IGNC,IGHC(3))
C-----YYDDDD, HHHHSS AND VALID-TIME (IF APPLICABLE) FOR GRID
EQUIVALENCE (IGDAY,IGHC(4)),(IGTIME,IGHC(5)),(IGTIMV,IGHC(6))
C-----DESCRIPTION OF GRICED VARIABLE (IN MC-FILE TERMS):
C----- NAME, SCALE, AND UNITS
EQUIVALENCE (IGVAP1,IGHC(7)),(IGVSPA,IGHC(8)),(IGWNP1,IGHC(9))
C-----DESCRIPTION OF VERTICAL LEVEL: VALUE, SCALE, AND UNITS
EQUIVALENCE (IGLEV1,IGHC(10)),(IGLeea,IGHC(11)),(IGLAI,IGHC(12))
C-----GRIDDEC-VARIABLE TYPE: =1 (TIME CDF) 2 (TIME AVG) 4 (LEVEL CDF)
C---- 8 (LEVEL AVG) OR ANY SUM OF THE FCREGECING
EQUIVALENCE (IGVTP,IGHC(13))
C-----FOLLOWING USED IF PARAMETER IS A VERTICAL (LEVEL) CDF OR AVG
C----- (SAME SCALE AS IGLEVEL)
EQUIVALENCE (IGLIF1,IGHC(14))
C----FOLLOWING USED IF PARAMETER IS A TIME CDF OR AVG (HHMMSS)
EQUIVALENCE (IGTIF1,IGHC(15))
C-----GRID ORIGIN, TYPE (I.E. TYPE OF PROJECTION)
EQUIVALENCE (IGCOORD,IGHC(33)),(IGTYPE,IGHC(34))
C-----SEQUENT CCCRCS (ICLAMPX,ICLCMPX,ICLAMX,ICLMX,ICINCR1)
C----- IFCPLC,ICLFC,LIPGLO,IGLCN) ALL HAVE 4 IMPLIEC DEG. PLACES.
C)-- LAT GOES FROM -500000 TO 500000, LNG GOES FROM -1800000
C)-- TO 1800000 (WEST IS+)
C-----TYPE 1 GRGS ARE PSEUDO-MEPCTOR
EQUIVALENCE (ICLAMPX,IGHC(35)),(ICLCMPX,IGHC(36)),(ICLAMX,IGHC(37))
,(ICLMX,IGHC(38)),(ICINCR1,IGHC(39))
C----TYPE 2 GRIDS ARE FCLAR-STEREOCYRAPHIC
C----GIVE ROW # OF NORTH POLE, COL # OF N.P., COL SPACING AT 60 DEG.
C----LONGITUDE PARALLEL TO COLUMNS (CEG)
C----EQUIVALENCE (IGFLG,IGCH(35)),(IGFSLG,IGCH(36)),(IGFSLG,IGCH(37)),
C----,(IGCLOF,IGCH(38))
C----INITIALS OF USER AND PROJECT # UNDER WHICH GRID CREATED
C----EQUIVALENCE (IGLSER*IGCH(41)),(IGPCC,IGCH(42))
C----CHARACTER ID SLAPPED ON PROGRAM (ARBITRARY)
C----EQUIVALENCE (IGID,IGCH(43))
C----EQUIVALENCE (RCN(1),IREC(1))
C----EQUIVALENCE (CCL,IREC(4))
DATA OFUF/20**/
DATA IFPRES/1000,850,700,500,400,300,250,200,150,100,70,50,30,0,
10,1000,850,700,500,400,300,250,200,150,100,70,50,30,0/
C----FOLLOWING 2 SPECIAL CASES FOR CALLAN'S ANMRC AND ECHR FIASCCS
DATA LCHAR/15**/ TEMP*6** DPT*6** HGT*/
DATA ICHAR/15**/ TEMP*6** TDPT*6** HGT*/
DATA KCHAR/15**/ TEMP*6** TDPT*6** HGT*1*
DATA JCHAR/15**/ TEMP*6** TDPT*6** HGT*/
DATA IGOT/22 0/
DATA MIGS/280800080/
DATA LUN/20*/LEK/100/
DATA IBIAS 0/
C
C
C
NGE=IFP(1,C)
IF (NGE.EQ.0)GO TO 500
ITERM=LUCE(-20)
CALL DCPEK('VASTEXT',LUN.LEN)
CALL DREAD(LLA,ITERM,ICCC)
NGFG=ICCC(42)
100 NGE=NGE+70
L=0
DO 10 N=NGE,NGE
ICK=IGGET(NGFG,LLA,32CC,IGRID(1),NR,NC,IGHCT)
C----FOLLOWING NECESSARY BECAUSE OF GRIDS WITH HCLLES
ITVNAME=IGHCT(7)
ITLEVEL=IGHCT(10)
IF (ICK.EQ.0)GO TO 2
IF (ICK.LT.0)GO TO 930
GO TO 10
2 CONTINUE
NPT=NR NC
DO 3 I=1,22
JE=I
IF (LCHAR(I),ECLI*ITVNAME)CC TC 4
100 IF (ICK-I.EQ.IGCH(1))GO TO 4
100 IF (ICK-I.EQ.IGCH(2))GO TO 4
100 IF (ICK-I.EQ.IGCH(3))GO TO 4
100 IF (ICK-I.EQ.IGCH(4))GO TO 4
3 CONTINUE
GO TO 10
DO 5 J=JE+22
10 IF (IGCT(J).EQ.1) GC TO 5
11 L=J
12 IF (IPRESS(.EQ.1).AND.ITLEVL) GC TO 6
13 5 CONTINUE
14 GC TO 10
15 6 CONTINUE
16 DO PO I=1,64
17 80 IGHE(I)=IGFCT(I)
18 KOUT(1)=6
19 KOUT(2)=IGDAY
20 KOUT(3)=IGTIME
21 KOUT(4)=IGLEVEL
22 KOUT(5)=L
23 KOUT(6)=IGNR
24 KOUT(7)=IGAC
25 CALL OLTINT(KCLT)
26 IGOT(L)=1
27 CALL SET UP FOR GLESS UPDATE
28 CALL BMGECE(*A4,8X,K5,7X,K5,7X,K5,7X),OEUFR.OEUF,CHAR(,E),ITLEVEL
29 ,IBIAS)
30 CALL TO(CELF)
31 CALL JSIG(*ENVA,*OEUFR(1),OEUFR(4),OEUFR(7),CELF(10))
32 NGOT=0
33 DO 11 J=1,22
34 11 NGOT=NGCT+IGCT(J)
35 IF(NGOT.EQ.22) GC TO 12
36 10 CONTINUE
37 IF (L.EQ.0) GC TO 550
38 12 CONTINUE
39 CALL SDEST(*FECATING GUESS FOR DAY *,IGDAY)
40 CALL SDEST(* HOUR = *,IGTIME)
41 CALL SDEST(* VALID TIME = *,IGTIM)
42 RETURN
43 900 CALL SDEST(*FIRST GRID NO.0*)
44 RETURN
45 930 CALL SDEST(* NABLE TO CFEN GRID FILE NO. *,MCFG)
46 RETURN
47 950 CALL SDEST(* CANNOT FIND ONE SIMPLE UGLY GRID*,O)
48 RETURN
49 END
C ? PROGRAM TO EXAMINE RETRIEVAL MODFILES USING CURSOR ON TV IMAGE
C ? KEYIN: EXVA *KEYWORD
C OSLCN SEARCH RADIUS IN DEG LAT TO FIND REPORT
C SSEC/KCICAS USERS MANUAL - CHAP12
C COMMON /NAV/FLAT,FLON,FLCC,SENLC,SNIN,IL,IE,IRAS,IFIC,IHMS,JS,JD,JC
C DIMENSION MCC(64),ICUT(300),KOUT(10),NOUT(10)
C DIMENSION LIST(300),ISCL(300),ILCC(300),LCCS(300)
C DIMENSION ILV(8)
C CHARACTER & FILE
C COMMON /DCC/IDCC(112)
C DATA *FILE,LLN,LEN,**VASTEXT,*,20,100/
C DATA MISS,2000000000/
C DATA ILV/1,5,7,5,10,11,12,13/
C IHCL=IKWP(*HELP*,1,C)
C IF(IHEL.1EQ.LIT(*HELP*)) GO TO 270
C IFRM=LUC(-1)
C IRAS=LUC(-11)
C IPIC=LUC(-12)
C CALL TVSAT(IFRM,IRAS,IPIC,IL,IE,JS,JD,JD)
C READ IN COCCENTRATION
C IF (LUC(81).1EQ.0) GC TO 120
C ITERM=LUC(-20)
C CALL DOPEN(MFILE,LLN,LEN)
C CALL CREAC(LLN,TERM,ICCC)
C CALL DCLCSE(LLN)
C INITIALIZE NAVIGATION
C UC=JS 9000000000JD
C CALL NVINIT(EETAIN,BETDC,IBR,PTIME)
C GO TO 122
C 120 CALL TSNIC(1,1,1,1,1,1,1,1,ICCC)
C 122 MDC=IDCC(40)
C MCR=IDCC(41)
C MUST CALL MCFEN TO USE MDGCT
C IF(MDC(MGC=1).1EQ.0) GO TO 900
C READ IN KEYS
C MEKEYS=MKEYS(MDC,-1,LIST,ISCL,ILCC,LCCS)
C SET UP INDICES
C DO 11 N=1,NKEYS
C IF (LIST(N).1EQ.LIT(*LAT*)) NLAT=N
C IF (LIST(N).1EQ.LIT(*LON*)) NLCN=N
C IF (LIST(N).1EQ.LIT(*T*)) GO TO 11
C NDX=N
C CC TO 12
C 11 CONTINUE
12 CONTINUE
13  ICK=MINFO( MENO, MDC )
14  IF ( ICK.LT.0 ) GC TO 500
15  ICK=MDCET( MEC, MDF, D, ICLUT )
16  MREC=ICUT(3)
17  SLCP=CKLF(*SLCP,1,0,75)
18  M=IPP(1,0)
19  M=IPP(2,0)
20  IF ( M .EQ. 0 ) M=N
21  ICUR=0
22  IF ( N .EQ. 0 ) GC TO 160
23  130 IF ( N .GT. MREC) GO TO 280
24  M=N
25  ICK=MDCET( MEC, MDF, AN, ICLUT )
26  FLATR=.0001 ICUT(NA)
27  FLONR=.0001 ICUT(NL)
28  IF ( ICK.LT.0 ) GC TO 200
29  GO TO 210
30
31  160 CONTINUE
32  ICUR=1
33  IF ( LUC(81).EQ.0 ) GC TO 162
34  FLIN=FLOAT(IL)
35  FELE=FLOAT(IE)
36  CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLON,1,NAV,ETAIN,BETDOT,0.)
37  GC TO 164
38  162 CALL TSNIC(C+1,IL,IE,1,ILAT)
39  CALL TSNIC(C+2,IL,IE,1,ILCN)
40  FLAT=.01 FLAC(ILAT)
41  FLON=.01 FLOAT(ILCN)
42
43  164 CONTINUE
44  C CONVET TO 4-E CONVENTION
45  FLON=FLCN
46  C NAV COMPLETE
47  DC 200 N=1,MREC
48  ICK=MDCET( MEC, MDF, A, ICLUT )
49  IF ( IOK.LT.0 ) GC TO 200
50  FLATR=.0001 FLAC(ICUT(NA))
51  FLONR=.0001 FLAC(ICUT(NL))
52  FCK=ABS(FLAT-FLATR)+ABS(FLON-FLONR)
53  IF ( FCK.GT.SLCP) GC TO 200
54  N=N
55  GO TO 210
56
57  200 CONTINUE
58  CALL SEEST(* AC SCULING AT THIS LOCATION ...*,C)
59  GC TO 250
60  210 CALL SEEST(* LAT LCN NMER (RE-EXAMINE)*,C)
61  KOUT(1)=3
62  KOUT(2)=FLAT-100.
63  KOUT(3)=FLCNF-100.
64  KOUT(4)=N
65  CALL CORTIN(KOUT)
66  CALL SEEST(* TSFC T850 T700 T500 T400 T300
250 T2C0.*,0)
106 KOUT(1)=8
107 NCUT(1)=8
108 L=1
109 Go 220 K=1.8
110 L=L+1
111 MM=(ILV(K)-1)*8+MCX
112 KOUT(L)=999
113 NCUT(L)=999
114 IF (IOLT(MM).EG.*MISS) Go To 220
115 KOUT(L)=IQLT(MM)
116 *OUT(L)=ICLT(MM)-ICUT(MM+5)
117 Continue
118 CALL COUTINT(KOUT)
119 CALL CLOTINT(NCUT)
120 IF(ICUR.NE.0) Go To 290
121 IF(M.EQ.0) Go To 290
122 ! =N+1
123 IF(M.GT.N) Go To 290
124 Go To 130
125 Call SEEST(* (NUN-STRT) (NUN-STCF)**,0)
126 CALL SEEST(* IF 'N' NUN, PICK UP RPT AT CURSOR LOCATION**,0)
127 Go To 290
128 Call SEEST(* NO SCALING AT RECORD NUMBER**,N)
129 290 RETURN
130 900 CALL SEEST(* CANACT OPEN MD FILE **USE MDU SET**,0)
131 RETURN
132 END
C ROUTINE TO PICK UP DATA ACCORDING TO CURSOR POSITION
C AND SOUND AREA POINT (SET BY VPVA)
C
C PARAMETERS ARE:
C (LH/LV/S/N) DEFAULT E) (LINE) (ELEMENT) (DIAGNOSTIC)
C BRIGHTNESS T; RADIANCE; FILTER; SPIN BUDGET; NAVIGATION
C
C SSEC/MCICAS USERS MANUAL - CHAP 12
C
DIMENSION IELF(400),VDAT(13),MIN(10),KCUT(10)
DIMENSION KCLT(10),KACUT(10),KOUT(10)
COMMON /ELIS/LISDAT
COMMON /LAST/LASLIN,LASELE,LELEV,ICHAR
COMMON /MCCE/IFIL(13),ISP(13)
COMMON /NAV/FLAT,FLCK,ZENLOC,ZEN,IL,IE,IRAS,IFIC,IMHS,JC,JD.
COMMON /RADC/RAD(13)
COMMON /SFEC/NGC(4)
EQUIVALENCE (KCLT(1),KACUT(1),KOUT(1),KOUT(1),KOUT(1),KOUT(1),KOUT(1),KOUT(1),KOUT(1),KOUT(1),KOUT(1))
DIMENSION SCALE(12)
DATA SCALE/E 1.0000,1.0000,1.0000,2.0000,3.0000,4.0000,5.0000,6.0000,7.0000,8.0000,9.0000,10.0000.
CALL I0(MIN)
IF (MIN(1),MO.E.LIT("HEL")) CR.MIN(1),MO.E.LIT("HEL") GC TO 190
LISDAT=MIN(4)
IP=1
IF (MIN(1),MO.E.LIT("F")) IP=2
IF (MIN(1),MO.E.LIT("S")) IP=3
34 IL=MIN(2)
35 IE=MIN(3)
36 IF (MIN(2),ME.E.0) GC TO 110
37 IFRAME=LCC(-1)
38 IRAS=LCC(-11)
39 IFIC=LCC(-12)
40 CALL TVSAT(IFRAME,IRAS,IFIC,IL,IE,IS,JC,JD)
41 JD=IS-100000+JD
42 IF (IP,ME.E.1) GC TO 130
43 IF (MIN(1),MO.E.LIT("E")) CR.MIN(1),MO.E.LIT("R") GC TO 130
44 IF (MIN(1),MO.E.LIT("N")) VDAT(1)=-1.
45 CALL VASDAT(IL,IE,VDAT)
46 CALL SDEST("" DAY = "JD"
47 CALL SDEST("" LINE ELEM LAT LCN LZN SZE
48 LEV SURF",C)
49 KCLT(1)=F
50 KCLT(2)=IL
51 KOUT(3)=IE
52 KOUT(4)=IRCLAD(FLAT 100.)
KCLT(5)=IRCLAC(FLAG=100)
KCLT(6)=IRCLAC(FLAG=100)
KCLT(7)=IRCLAC(FLAG=100)
LATT=KCLT(4)
LCMN=-KCLT(5)
CALL HFTFCF(LATT,LCMN,LELEV,ICHAR)
KCLT(3)=LELEV
ISURF=0
IF (ICHAR.NE.0) ISURF=1
KCLT(9)=ISURF
CALL CLINT(KCLT)
IF ("I".EQ."LIT(*N*")) GO TO 140
RETURN
130 CONTINUE
C PICK UP DATA WITHOUT NAVIGATION
JD=0
JF=1
CALL VASDAM(IL*,IE*,VEAT)
140 CONTINUE
CALL SDEST(CHAN1,CHAN2,CHAN3,CHAN4,CHAN5,CHAN6,CHAN7,CHAN8)
SC=100
DO 160 K=1,8
KCLT(K+1)=VEAT(K)
IF (MIN(1).NE.LIT(*R*)) GO TO 150
VEAT(K)=RAC(K)
SC=SCALE(K)
150 IF (VEAT(K).NE.55555) KCLT(K+1)=VEAT(K)*SC
KCLT(K+1)=ISFA(K)
KCLT(K+1)=ISFA(K)
160 CONTINUE
KCLT(1,IF)=8
CALL CLINT(KKCLT(1,IF))
CALL SDEST(CHAN9,CHAN10,CHAN11,CHAN12,*,0)
DO 180 K=1,4
KCLT(K+1)=VEAT(K+4)
IF (MIN(1).NE.LIT(*R*)) GO TO 170
VEAT(K+F)=RAC(K+F)
SC=SCALE(K+F)
170 IF (VEAT(K+F).NE.55555) KCLT(K+F)=VEAT(K+F)*SC
KCLT(K+F)=ISFA(K+F)
KCLT(K+F)=ISFA(K+F)
180 CONTINUE
KKCLT(1,IF)=4
CALL CLINT(KKCLT(1,IF))
RETURN
150 CONTINUE
CALL SDEST("E/R/F/S/N...DEFAULT=B") (LINE) (ELEM) (VASDAM DIAGNOS
IC DISPLAY*)
C CALL SDEST("... N (NAV) GIVES SFC DATA IF AVAILABLE... RTVL=FILE PR
102 C MUST BE SET")
RETURN
104 END
SUBROUTINE MAIN
C ? VARS RADIANCE FRCGRAPH USES KEYWORDS OR *VASTEXT* DEFAULTS
C ? KEYIN: SRA C <ACTION>
C ? POSITIONAL PARAMETER:
C ? ACTION: *CLE* TO CLEAR KEYWORD LIST
C ? *GC* TO BEGIN MAKING RETRIEVALS
C ? KEYWORDS:
C ? TYP=F (FCRCE) N (KSTAR) C (CLEAP) C (CVCST) B (DEFAULT)
C ? SFC=IL IE (LINE AND ELEMENT SPACING OF RETRIEVALS (10*10))
C ? SIZ=N (FCV AVERAGED BOX SYMMETRIC (5*5))
C ? SFC = (NC SURFACE ANALYSIS OPTION) (DEFAULT=0)
C ? ENS=LL LE (LAST LINE AND ELEMENT TO DEFINE AREA (VASTEXT))
C ? BEG=IL IE (FIRST LINE AND ELEMENT TO DEFINE AREA (CURSCR))
C ? NTERM= (TERMINAL NUMBER TO DEFINE *VASTEXT* (LOCAL))
C ? BUG=1 (FOR CEEUUG DIAGNOSTICS (C))
C ? PLT=1 (PLCT CN GRAPHICS (0))
C ? ICE=1 (MAKE CALIBRATION RLN USING GUESS)
C ? ALT=1 (TAKE IL IE LL LE FROM VASTEXT)
C ? OLC=1 DATA IS BEFORE JAN 15 1982
C ? NCFC=1 DO NOT USE THE KSTAR OPTION
DIMENSION IFILN(2),KCLT(10),ICU(3),ICT(6),IILP(3)
VDAT(13),BTC(13)
NSAM(13),AVG(13),RRAS(121,13),NSRAS(121)
PF(64),LELF(13)
DIMENSION LC(13),MFSIN(13),ICZ(15),IUSE(13),CALS(6),FUM(3),TC(40)
DIMENSION KLS(12),TSAT(40),IRET(246),IREAC(246),ER(12)
DIMENSION TES(40),LEC(20),LCHR(20),CGS(11)
DIMENSION FST(15),TST(15),TST(15),SPD(15),CIR(15),STABIL(12)
CHARACTER 12 CGES,CFP,CKLPC,CTYP,CTYFS
COMMOM /DANGLE/FLAT,PLCN
COMMOM/SDCC/SDCC(6)
COMMOM/CCIC/CCIC(100)
COMMOM/TERM/TERM
COMMOM/AUTC/ATMCX/ILU,ILL,IEL,IER,MAG
COMMOM/SIZE/AEXS
COMMOM/SFC/FS,FR,DEL7,DEL8
COMMOM/ATMCS/F(40),T(40),W(40)
COMMOM/ARENT/ICIR(64)
COMMOM/GLESS/TGES(15),DGES(6)
COMMOM/LAST/LASLIX,LESE,LELEV,ICHR
COMMOM/RADV/SRAC(13)
COMMOM/SURF/IS10,ITSC,FSC,IPSTA,IELEV,LSTA
COMMOM/AVAT/LCAL,VLCA,VZEN,SEZ,IL,IE,IRAS,IFIC,ITIME,JTIME,JDAY
COMMOM/MECE/ICET(13),ISPIN(13)
COMMOM/CCE/CV(12),CV(12),EV(12)
COMMOM/USE/LICLH(12,2)
COMMOM/DELG/KBUG
COMMOM/JAN/ICLD
C IF ICDD NE C WE USE THE FIXED DELTA F PATHER THAN LINE
C DOCUMENTATION (DATA BEFORE 15 JAN 82)
C EQUIVALENCE (IRET(17),IRET(1))
C EQUIVALENCE TO IRET(ROWLEN-1)
C ABOVE TO FACILITATE CHANGING LENGTH OF ROW HEADER
C DATA CIR/15 C/,SFC/15 Z0/
C DATA IRET/246 28080C8C0/NSIZE/245/
C DATA LMC/C,40,35,35,37,36,35,34,31,28,25,24,23,20,
C 16,15,14,13,11/
C DATA LCHR/SFC *,1000 *,950 *,920 *,850 *,800 *,700 *
C *,670 *,650 *,640 *,630 *,620 *,600 *,600 *,590 *
C *,490 *,390 *,290 *,200 *,150 *,100 *
C DATA ICT/1,2,3,4,5,6/
C DATA ICL/7,8,9,10/
C DATA ICL/P/
C DATA ITCH/E/
C DATA RCX/25,2658,GR_AV/980,665,TCTC/347/
C DATA WISG/555959,*,WISG/Z80808080/
C DATA NL/40,ASMAB/121,*MAXRET/699/
C DATA ILINE/C,IELEM/0/
C
C FOLLOWING IS PROGRAM VERSION CATE ... KEEP CURRENT!!!
C DATA IVER/84023/
C CALL CALDAY(IVER,IVY,IVY,IVC,IVC)
C CALL ENKODE(*(132),T1,*EENIN SRAC, VERSION OF *,
C I2,1X,4,12)*,LEU,F,IVC,IVM,IVV)
C
C READ IN VASTEXT AND ROW HEADER
C CALL VRTIC(IRET,0,0)
C CALL MDNAME(ICCC(40),IFILN)
C ABOVE SETS OF FILE NAME TO FORCE OUTPUT AT ENO OF EACH LINE
C IF (CPP(1,*) NE ,CLE*) GO TO 100
C DO 50 I=50,60
C 50 IDCC(I)=0
C IDCC(52)=10
C IDCC(53)=5
C CALL VRTIC(IRET,0,1)
C RETURN
C 100 CONTINUE
C LLNW=IDOC(25)
C LS=IDOC(26)
C LN=LLNW/1000
C LS=LLNW/1000
C LE=MOD(LLNW,1000)
C 100 C ABOVE WILL LIMIT AREA OF RETRIEVALS
C 101 C TEST FOR CALIBRATION RUN
C 102 C ICE=IKWP(*ICE*,1,0)
C 103 C ITYP=IDOC(50)
C CTYP=IKWP(*TYF*,1,-)

444
IF (CTYP. NE. ' ') GO TO 105
IF (ITYP. EQ. 21) CTYP = ' C'
IF (ITYP. EQ. 22) CTYP = ' N'
IF (ITYP. EQ. 23) CTYP = ' F'
FCRCE = 0.
IF (CTYP. EQ. ' F ') FCRCE = 10.
IF (CTYP. EQ. ' C ') FCRCE = 10.
IF (CTYP. EQ. ' N ') FCRCE = 10.
C *F_CRCE* USEC AS NCA ZERO FLAG TO SKIP GPCSS ERRCR CHECKS
CONTINUE
ITYF = 0
IF (CTYP . EQ. ' C ') ITYP = 21
IF (CTYP . EQ. ' N ') ITYP = 22
IF (CTYP . EQ. ' F ') ITYP = 23
IDCC (50) = ITYP
IDEF = IDCC (54)
NCSCF = IKWP (' SFC ', 1, ICEF)
IDCC (54) = NCSCF
ICEF = IDCC (52)
INCR = IKWP (' SFC ', 1, IDEF)
IDCC (52) = INCR
INCR = IKWP (' SFC ', 2, INCR)
IDEF = IDCC (53)
NBXS = IKWP (' SIZ ', 1, ICEF)
IF (MOD (NBXS * 2, EQ. 0) NBXS = NBXS - 1
IF (NBXS . GT. 11) NBXS = 11
C MAX SIZE FOR RETRIEVAL IS 11 11
IF (NBXS . LT. 3) NBXS = 3
IDCC (53) = NBXS
NSMAX = NBXS NBXS
ICLC = IKWP (' LCL ', 1, 0)
KBUG = IKWP (' ELG ', 1, 0)
IF (KBUG . NE. 0) CALL TQSET (KBUG)
IDEF = ITERM
C USE KEYWCRF FOR TERM WHEN RUNNING FROM BACKGRCLND
ITERM = IKWP (' ITERM ', 1, ICEF)
IDCC (59) = ITERM
IDCC = IDCC (60)
IFLT = IKWP (' IFLT ', 1, IDEF)
IDCC (60) = IFLT
NLP1 = NLP + 1
C SET INDICATOR OR SQUARE SAMPLES (IN VASEAT PRCESSING)
IBOX = 1
C SET UP PRCESSING DAY FOR DOCUMENTATION IN PCW HEADER
CALL GETDAY (ICPRCS)
MDO = IDCC (38)
MNO = IDCC (40)
LASRT = IRET (3)
IFRM = LEC (1)
INRAS = LUC (- 1)
INPIC = LUC (- 12)
C SEE IF THIS IS AN ALTO MODE RUN
IF (IKWP(*ALTIC-,1,.0).EQ.0) GO TO 112.
112 CONTINUE
LLINE=IKWP(*END-,1,LLINE)
LELEM=IKWP(*END-,2,LELEM)
ILINE=IKWP(*EG-,1,ILINE)
IELEM=IKWP(*EG-,2,IELEM)
JTIME=IOCCC(2)
JDAY=IOCCC(1)
CALL VRTIC(IRET-.0,-1)
IF (CPP(1,*,*,NE.*GC*) RETURN
IF (ILINE IELEM NE.0) GO TO 218
CALL TVSAT(IFRM,INPAR,INPIC,ILINE,IELEM,JSAT,JDAY,JTIME)
C RESET JDAY FOR NAVIGATION WITHIN VASCAT
JDAY=JSAT 1CC0000+JDAY
IF (LLINE LELEM NE.0) GO TO 220
LLINE=ILINE
LELEM=IELEM
220 CONTINUE
VDATE(1)=-1.
ILINES=ILINE
IELEMS=IELEM
LASLIN=0
C VASCAT CHANGES ARGUMENTS *ILINES* AND *IELEMS*
CALL VASCAT(ILINES,IELEMS,VDATE)
VDATE(1)=VMIOSG
ISAT=ISATN(ICIR(3))
ILRES=ICIR(12)
IERS=IDIR(12)
C SET DEFAULT TO SQUARE SAMPLES..IE EQUATE RESOLUTIONS
INC=ILRES/IERES
INCIL=ILRES INCPF
IF (INCIL .EQ. C) INCIL=1
INCIE=IERES INCFE*INC
IF (INCIE .EQ. C) INCIE=1
CALL GETFRM(IFRM,WF)
MAG=MF(10)
MAG2=MAG/2
MAG=MAXC(MAG,6)
IF=INCIE/2
CTYPF=CTYFP
ITYPY=ITYPF
CC 2400 IL=ILINE,LLINE,INCIL
LASLIN=-1
C ABOVE IS NECESSARY FOR REPEATED CALLS TO VASCAT
LEEG=0
NOGO=0
IEF=IELEM-INCIE
C BEGIN IMPLICIT CC LCCP FOR LINE
C BACK UP HALF INCREMENT IF PREVIOUS ATTEMPT FAILED
216 IF (NOGO.NE.0) IEP=IEP-IDEL
217 CC 330 I=1*11
218 CC 280 N=1*?Max
219 280 RADES(N.*I)=VMISG
220 IE=IEP
221 RDAT(I)=VMISG
222 AVG(I)=0.
223 VDAT(I)=VMISG
224 IUSE(I)=0
225 NSPIN(I)=0
226 330 NSAM(I)=0
227 MSAM=0
228 SELEVE=0.
229 SLAT=0.
230 SLON=0.
231 IBCX=IBCXS
232 N=0

C *** BEGIN IMPLICIT LOOP TO OBTAIN ALL DATA FOR RETRIEVAL
235 C COMPLETION IS FLAGGED BY IBCX=3 SET IN VASCAT
236 305 N=N+1
237 VDAT(I)=C.
238 ILINES=IL
239 IELES=IE
240 CALL VASDAT(ILINES,IELES,VDAT)
241 IF(N.GT.1) GO TO 330
242 C CHECK CENTER LAT/LON/ZENITH
243 LA=VLAT
244 IF (LA.GT.LK.LC.LK.LAT.LAT.LS) GO TO 2400
245 TLOK=VLON
246 IF (TLON.LT.C) TLOK=360.*TLOK
247 LO=TLOK
248 IF (LC.GT.LK.LC) CC TC 2190
249 IF (LC.LT.LE) CC TC 2390
250 IF(ABS(VZEN).LE.60.) GO TO 330
251 IF (LBEG.IG.1) GO TC 2390
252 GO TO 2350
253 330 IF(VDAT(I).LT.0.) CC TC 420
254 IF (KBLG.IG.5)
255 CALL ENKCDR(*T2."W.LAT.LON.EL.CHAN 5-8 */3.2F7.2.IG.4F7.2*)
256 LEUF."N.PLAT.FLOC.EL.LAT.VDAT(5))
257 C FOLLOWING STATEMENTS ARE COMMENTED BECAUSE OF SMALL DETECTOR
258 CCCCCIF(VDAT(IW).*.EG.VMISG)CGOCTCC420CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
259 CCCCCDCC340Cl=1.3CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
260 CCCCCCIV=ICL(I)CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

447
CC560 CONTINUE

DO 400 I=1,13

IF(VDAT(I).LT.180.CR.VDAT(I).GT.330.) VDAT(I)=141560
IF(VDAT(I).GT.280) GC TO 400

* IF(IMF.EQ.0) GC TO 380

MSAM=MSAM+1

IMF=0

380 CONTINUE

RADS(N,I)=VDAT(I)

SAVE REPRESENTATIVE SPIN BLCC..WE ASSUME SPIN EULUGET DO NOT CHG

NSPIN(I)=ISFIN(I)

400 CONTINUE

SELEV=SELEV+LELEV

SLAT=SLAT+FLAT

IF (PLON.LT.0.) PLON=360.*PLON

SLON=SLON+FLON

420 CONTINUE

IF (ICHX.NE.3) GC TO 305

IF (MSAM.EQ.0) GC TO 2190

LBEG=1

FSAM=FLOAT(MSAM)

SELEV=SELEV/FSAM

SLAT=SLAT/FSAM

SLON=SLON/FSAM

IELEV=SELEV+.5

LAT=SLAT+100.*.5

LON=SLON+100.*.5

IVZ=IRCUND(VZEN 100.)

ISZ=IRCUNCI(SZEN 100.)

IF(KBUG.EQ.0) GC TO 430

CALL ENKCCE(*,T5,"LINE",4X,",ELEM",5X,"LAT",5X,"LON",4X,

"LZEN",4X,"SZEN",4X,"IELE"/1),LBUC)

CALL ENKCCE(*,T7E/),LBUC,IL,IE,LAT,LON,IVZ,ISZ,IELEV)

CALL SDEST(*,SAMPLE = *MSAM)

430 CONTINUE

IPSTA=VMSG

ITSTA=VMSG

IDSTA=VMSG

CALL SURGE(NCSFC)

TSTA=0.01 FLCAT(ITSCFC)

TMN=TSTA-10.

IF (NCSFC.NE.0) TMN=TMN-.5

GENERATE CLEAREST RADIANCES.

DO 580 J=1,NMAX

DO 560 K=1,12

IF(RADS(J,K).EQ.VMSG) GC TO 560

IUSE(K)=IUSE(K)+1

NSAM(K)=NSAM(K)+1
AVG(K) = AVG(K) * RADS(I*K)

560 CONTINUE

580 CONTINUE

LIM = 3

IF (IUSE(E) .LT. LIM) GC TO 2150

C INSIST ON RAW SAMPLE OF 3 OBS

DO 590 K = 1, 12

KUSE(K) = 1

IF (IUSE(K) .LT. LIM) IUSE(K) = 0

590 CONTINUE

IF (CTYP*EG*K) GC TO 770

C SEEK WARMEST WHEN SAMPLE IS ADEQUATE FOR ALL CHANNELS

TMAX = 0.

T6MAX = 0.

JSAV6 = 0

JSAV = 0

DO 600 J = 1, NSMAX

DO 595 K = 2, 10

IF (IUSE(K) .LT. EG*0) GC TO 600

595 CONTINUE

T6 = RADS(J*6)

IF (T6 .LT. T6MAX) GC TO 557

IF (T6 .LT. LT.T6MAX) GC TO 557

T6MAX = T6

JSAV6 = J

597 CONTINUE

TW = RADS(J*8)

IF (TW .LT. TWMAX) GC TO 600

IF (TW .LT. LT.TWMAX) GC TO 600

TWMAX = TW

JSAV = J

600 CONTINUE

IF (TWMAX .LT. TWMAX) GC TO 2150

IF (KBUG .NE. 0)

CALL ENKCCE(" (" JSAY = " +I3," + JSAY6 = " +I3") + LEUF,JSAY,JSAY6)

IF (JSAY .LT. JSAY6) GC TO 2190

IF (JSAY6 .LT. JSAY6) GC TO 2190

RLOC = RADS(JSAV*8) - 2.0

IF (NOSFC .LT. EG*C) RLOC = AMIN1(RLOC,TSTA)

RLOC6 = RADS(JSAV6*8) - 1.5

RLOC9 = RLOC - 20.

RLOC5 = RADS(JSAV5*8) - 1.0

RLOC4 = RADS(JSAV4*8) - 1.0

DC 650 K = 3, 12

NSAM(K) = 0

650 AVG(K) = 0.

DC 720 I = 1, NSMAX

DO 660 I = 3, 12

660 LC(J) = 1

IF (RADS(I,E) .LE. VMISG .AND. RADS(I,E) .GE. RLC6) GC TO 670

LC(E) = 0

LC(12) = 0
365 IF(RADS(I,6) .NE. VMISG.AND.RADS(I,6) .GE.RLC9) GC TO 670
366 LCI(9) = 0
367 LC(10) = 0
368 670 CONTINUE
369 IF(RADS(I,8) .NE. VMISG.AND.RADS(I,8) .GE.RLC8) GC TO 680
370 LCI(8) = 0
371 LC(7) = 0
372 LC(12) = 0
373 LCI(5) = 0
374 LC(6) = 0
375 IF(RADS(I,5) .NE. VMISG.AND.RADS(I,5) .GE.RLC5) GC TO 680
376 LCI(4) = 0
377 LC(9) = 0
378 IF(RADS(I,4) .NE. VMISG.AND.RADS(I,4) .GE.RLC4) GC TO 680
379 LCI(3) = 0
380 LC(10) = 0
381 680 DC 700 J=3,12
382 IF(LCI(J) .EQ. 0.C.R.RADS(I,J) .GE. VMISG) GC TO 700
383 NSAM(J) = NSAM(J) + 1
384 AVG(J) = AVG(J) + RADS(I,J)
385 700 CONTINUE
386 720 CONTINUE
387 740 CONTINUE
388 760 DC 760 I=1,12
389 VDAT(I) = VMISG
390 SRA(I) = VMISG
391 IF(NSAM(I) .GE. 0.C.R.IS(F(I) .GE. 0) GO TO 760
392 VDAT(I) = AVG(I) / NSAM(I)
393 760 CONTINUE
394 C DUE TO SUBPCE 1 WE HAVE TO CHECK ON CHANNEL 7 SAMPLE INSTEAD
395 C OF THE WINDO CHANNEL
396 C
397 C
398 C
399 C
400 C
401 C EXERCISE *N* CCTION
402 770 DC 780 I=1,12
403 780 IUTCH(I,1) = IULSe(I)
404 C CHECK FOR KC NSTAR CCTION FOR KC
405 IF(IKW(*) .NE. 0) GO TO 2100
406 CALL NSTAR(RADS, RAS, VDAT, IFAIL, KSNAP)
407 IF(IFAIL .GE. 0) GO TO 760
408 IF(KFWC .GE. 0) CALL SEEST(* NSTAR FAILURE NO. */ IFAIL)
409 GO TO 2190
410 790 CONTINUE
411 IF(KFWC .GE. 6) GC TO 800
412 CALL ENKCDE(*72, "BRIGHT-TEMP", 12FP.2, 7BLF, VDAT)
413 800 CONTINUE
414 DC 830 I=1,12
415 EX(I) = 100.
416 IF(VDAT(I) .GE. VMISG) GO TO 830
450
SRAD(I)=VELA*C(VCAT(I),1)
DBCE=VCF3TE(VCAT(I),1)
IF(SPIN.EG.C.OR.DBDE.EG.C) GO TC 830
EX(I)=EV(I)/CBCE/SQRT(SPIN)
IF(I.EC.1) EX(I)-0.5*EX(I)
IF(I.EC.5) EX(I)-2.0*EX(I)
830 IF(EX(I).EQ.100.) IILSE(I)=0
C DATA ACQUISITION COMPLETE
TS=VSK*INT(VCAT,.0,.CUM)
IF(TS.GT.310.) TS=2.*(VCAT(I)-VCAT(7))
NOC=0
CC TO 2200
2190 NOG=1
CC TO 2380
2200 CONTINUE
C PREPARE CUTFLT BUFFER
DC 2201 K=1,N SIZE
2201 IRET(K)=MISC
IRET(4)=LUC(-17)
IRET(5)=IVER
IRET(6)=INCRE
IRET(7)=INCRL
IGD=MOC(ICCC(33)+1000)
IRET(8)=ICCC(24)+IGD+10000
IRET(9)=ICCC(6)
CC 2202 K=1,5
2202 IRET(F+K)=ICCC(K)
IRET(15)=0
IRET(16)=ICFROS
C FILL IN USER MCD FLAG
IRETD(3)=-555
IRETD(4)=LASRET+1
IRETD(5)=LAT+100
IRETD(6)=LCR+100
IRETD(7)=ICCC(2)
IRETD(8)=MSAM
DC 2203 N=1,12
IF(IUSE(N).EQ.0) GC TO 2203
IRETD(I+E+N)=1CC*VCAT(N)+0.5
IRETD(21+N)=EX(N)+10000*
CONTINUE
IRETD(36)=1CC*TS
IRETD(39)=SZEK-100.
IRETD(40)=VZEN-100.
2218 ITYF=21
IF(I.FlG.EC.C) ITYF=23
IF(CTYF.EC.K) ITYF=22
IRETD(50)=ITYF
IRETD(57)=IELEV
C MD CUTFLT BUFFER COMPLETE
LASRET=LASRET+1
CALL VRTIC (IRET, LASRET, 1)
IF (IPLT, EC, C) GC TO 2360
CALL SATTV (IFRM, IL, IE, IRAS, IPIC, USAT, JDAY, UTIME).
JDAY = JSAT + JDAY
KOLOR = 2
IF (CTYF, EC, **N**) KCLR = 3
IF (IFLG, EC, C) KCLR = 1
ITEM = VDAT (8)
CALL VASCIC (IRAS, MAG2, IPIC, MAG2, ITEM, MAG, 1, KCLR)
2360 CONTINUE
IF (LASRET, EC, MAXRET) GC TO 2390
2380 CONTINUE
GO TO 230
2385 CONTINUE
WRITE ROW HEADER AND VASTEXT
2390 IRET (1) = MCC (JDAY, 100000)
IRET (2) = MCC (EC)
IRET (3) = LASRET
CALL VRTIC (IRET, 0, 1)
2400 CONTINUE
IF (LASRET, EC, MAXRET) GC TO 2500
CALL SDEST (** NC, CF SNDGS, IN RTVL, FILE = **, LASRET)
RETURN
END
C ? VAS RETRIEVAL PROGRAM; USES MFILE DRIVER *VASTEXT* DEFAULTS
C ? KEYIN: SRET N1 N2 <KEYWORDS>
C ? POSITIONAL PARAMETER:
  N1 FIRST RETRIEVAL IN FILE (DEFAULT TO 1)
  N2 LAST RETRIEVAL IN FILE (DEFAULT TO VASTEXT MRET)
C ? KEYWORDS:
C ?
C ? TYPE=F (FCRCE)
C ? CSS=C (CLIM) P (PROFILE) G (GRID=DEFAULT)
C ? SPC=1 (KC SURFACE ANALYSIS OPTION) (DEFAULT=0)
C ? TCR=1 (TERMINAL NUMBER TO DEFINE *VASTEXT* (LOCAL))
C ? BUC=1 (FCR DEBUG DIAGNOSTICS (0))
C ? PLT=1 (PLOT ON GRAPHICS (0))
C ? ENH=1 (TURN OFF ENHANCEMENT)
C ? ICE=1 (MAKE CALIBRATION RLN USING GUESS)
C ? ITR=1 (MAX NO OF ITERATIONS) (DEFAULT=3)
C ?
C ? IF NCR-ZERO USE BIAS VECTOR IN VASTEXT
C ?
C ? DIMENSION !FILNM(4),KOUT(10),ICUR(3),ICT(6),IHLK(3)
C ? ,F(40),E(40),L(40),TAU(40),CELT(6),VCA(13),BCAT(13)
C ? ,MF(64)
C ? ,LEUC(33),TALS(40,6),TAUW(40,4),LG(40,6)
C ? DIMENSION LC(13),ILSE(13),EELS(6),HUM(3),TC(40),RRHC(3),RHC(3)
C ? DIMENSION KLSE(13),TAŚV(40),IRE(246),IRETD(246),EX(13)
C ? DIMENSION TGS(40),LRC(20),LCHR(20),CES(11)
C ? DIMENSION PST(15),ST(15),TSDT(15),SPD(15),DIR(15),STABIL(12)
C ? CHARACTER 12 CGES,C,CKJF,C
C ? COMMON/DLRG/KEUG
C ? COMMON/SDFC/ISDFC(E)
C ? COMMON /ESTEB/TBE(12),TRN(40,12)
C ? COMMON/DCC/ICCC(100)
C ? COMMON/TERM/ITERM
C ? COMMON/SFCS/PWR,DEL7,DEL8
C ? COMMON/ATMS/E(40),T(40),W(40)
C ? COMMON/GUESS/TGES(15),FGES(6)
C ? COMMON/RACV/VRAC(13)
C ? COMMON/SURF/IZ10,ITSFCC,ICSFC,IPST,IELEV,LSTA
C ? COMMON/NAV/VLAT,VLCK,VZEN,SZEN,IEL,IRAS,IPIC,ITIME,TIME,CAY
C ? COMMON/OCCE/ICET(13),ISPIM(13)
C ? COMMON/GCA/CH(12),CW(12),EV(12)
C ? COMMON/LSEC/ILC(12,2)
C ? EQUIVALENCE (IRET(17),IRETD(1))
C ? EQUIVALENCE (TALS(1,1),TRN(1,1)),(TAW(1,1),TRN(1,7))
C ?
C ? ABOVE TO FACILITATE CHANGING LENGTH OF RCW HEADER
C ? DATA DIR/15 C.,/SPD/15 0.
FOLLOWING IS PROGRAM VERSION DATE ... KEEP CURRENT!!!
CONTINUE
PREF=CW(1,1056)
C PREF IS USED TO BLEND VOLTAGE PROFILES WITH RETRIEVAL AT
C EACH ITERATION. BLENDING IS A LEVELS BELOW PREF
C
LCTRL=IKWP(*ELG',1.0)
IF (LCTRL) CALL TGSFT(LCTRL)
C GET TERMINAL NO. THIS HAS ALREADY BEEN LOADED IN VRTIC. OVERRIDE WITH
C VASTEXT IF THAT VALUE NOT ZERO
IDC=IDCC(65)
IF (IDC) IDC=TERM
TERM=IKWP(*TER',1,IDC)
ICCC(65)=TERM
KPLG=LCTRL
IDC=IDCC(60)
IPLT=IKWP(*PLT',1,IDC)
ICCC(60)=IPLT
IF (IPLT) CC TO 120
C INITIALIZE NAVIGATION
IRAS=LUC(-11)
IPIC=LUC(-12)
IFRM=LUC(-1)
CALL TYSAT(IFRM,IRAS,IPIC,IL,IE,JS,JDAY,JTIME)
JDAY=JS 100000+JDAY
CALL NVINIT(BETAIN, PETDOC, INAV, PTIME)
C GETFRM (IFRM, PF)
MAG=MF(10)
MAG=MAG(10)
MAG2=MAG/2
LUN=NL+1
IB=1
C SET UP PROCESSING DAY FOR DOCUMENTATION IN ROW HEADER
CALL GETDAY(IDCPRCS)
MDG=IDCC(38)
MDR=IDCC(40)
LASRET=IRET(3)
ENH=IKWP(*ENH',1,0)
CAY=IDCC(1)
TIME=IDCC(2)
IB=IP(1.1)
LASRET=IPP(2,LASRET)
RET=0
OC 2400 NA=NE*LASRET
CALL VRTIC(IRET,NN,0)
C LOCK AT USER MOD FLAG; LT C MEANS IT HASN'T BEEN RETRIEVED
C EG C MEANS IT HAS BEEN RETRIEVED
C GT 0 MEANS IT'S EITHER FAILED OR BEEN EDITED
IF (IRET) CC TO 2400
C IF (KPLG) CALL SDEST(* BEGAN SMDG NO. *,NN)
OC 2700 N=1.12
IUSE(N)=0
VDATE(N)=9592
C IF (IRET) CC TO 700
700 CONTINUE

156 C THERE ARE TWO SETS OF "USE" FLAGS IN THIS PROGRAM
157 C IUSE IS SET TO 0 IF DATA IS MISSING OR CHANNEL IS INTENTIONALLY
158 C SET ASIDE. KUSE IS SET TO 0 IF CHANNEL IS BELIEVED TO BE
159 C CLOUD CONTAMINATED; IN THIS CASE GUESS PROFILE IS IMPLICITLY
160 C BELIEVED UNLESS IT IS A NO SURFACE CLIMATOLOGY CASE.
161 IF (KUSE.EQ.0)
162 CALL ENKCDE( *(T2","BRIGTH-TEMP", 12F7.2/) ) , LBFL , VDAT)
163 IF (KUSE.EQ.0)
164 CALL ENKCE( *(T2,"CHAN FLG"," 12I7/) ) , LBFL , IUSE)
165 TSE=1.0E1 FLCAT(IRETD(36))
166 SZEN=1.0E1 FLCAT(IRETD(36))
167 VZEN=1.0E1 FLCAT(IRETD(40))
168 VLAT=0.001 FLCAT(IRETD(5))
169 VLEN=0.001 FLCAT(IRETD(6))
170 IYP=IRETD(50)
171 IELEV=IRETD(67)
172 C ZERO OUT PREVIOUSLY RETRIEVED QUANTITIES
173 IRETD(35)*=0
174 DC 380 I=41 , NSIZE
175 IRETD(1)=0
176 C RESTORE TYPE AND ELEVATION
177 IRETD(60)=IYP
178 IRETD(67)=IELEV
179 NCOC=1
180 CALL GFSFRC(IGES,NCSFC,PCDC)
181 DO (TGES(I).LT.0.0) GC TO 2200
182 PSTA=IPSTAA
183 TSTA=0.01 FLCAT(ITSCFC)
184 TC=2.0E1 VDAT(E)-VDAT(7)
185 C CHECK ON THE HOPELESSLY COLD , PRESUMED OVERCAST
186 IF ((TSTA+TCK).GT.15.) GC TO 2200
187 C LET HCT GROUND GC FOR ACW
188 C IF (ABS(TCK-TSTA).GT.15.) GC TO 2200
189 C FIND "IS" 1ST LEVEL BELOW SURFACE (PRESSURE)
190 IS=NL
191 DO 400 J=1,20
192 IP=NL(I)-J
193 DP=P(I)-PSTAA
194 IF (DP.GE.0.) GC TO 400
195 IS=I+1
196 GC TO 420
197 400 CONTINUE
198 420 CONTINUE
199 C BLEND GUESS WITH SURFACE DATA (IF THEY EXIST)
200 IF (NCSFC.NE.0.) GC TO 460
201 C BEGIN AT PCTE FOR NORMAL GUESS , ECTE FOR CLIMATOLOGY
ILC=37
IF (IGES.EC.EC.1) ILC=31
IF (ILC.EC.IS) GC TO 460
DTS=TSTA-T(IS)
DTUF=DTS/(F(TS)-P(ILC))
DC 440 I=ILC*IS
440 TI=E(T(I)+CTCF.P(I)-P(ILO))
CONTINUE
DO 425 T=1,40
DO 425 T=1,40
425 TSAVF(I)=T(I)
C PREPARE GUESS INFO FOR KDP FILE
DO 490 I=1,20
J=LMT(I)
480 TGS(I)=T(J)
TCS(I)=.91*FLCAT(ITSCFC)
DC 485 T=2,11
J=LMT(I)
DFT=DEPRT(F(J),T(J),W(J))
485 DGS(I)=DFT
DGS(I)=.91*FLCAT(IESCFC)
IF (KBUG.EQ.0) EC TO 540
CALL EMKCC(F((132),T6="PRESSURE",.15F7.1)/.LEUF.F(26))
CALL EMKCC(F(T6="GUESS TEMP",.15F7.1)/.LEUF.T(26))
CALL EMKCC(F(T6="GUESS WASH",.15F7.1)/.LEUF.W(26))
CALL EMKCC(F(T2="21000 =",.98*,TSFC="",.98*,TDSFC="",.98*)
*LEUF.IZ10.IFTSC.1ESCFC)
540 CONTINUE
TESTA=.91*FLCAT(IESFC)
EC=TSTA-TDSTA
PSTA=IPSTA
CALL WMIX(PSTA,TSTA,DD,IPSTA.1)
IF (CTYP.EC.*C) GC TO 665
C DETERMINE A TENTATIVE CLOUD FLAG (IFLC). THIS WILL BE USED
C TO ESTABLISH RH BOUNDARY AND IN T RETRIEVAL. IF FLAG IS SET
C TO NON-ZERO WE WILL NOT DO ENHANCEMENT AND WE WILL NOT
C BELIEVE SKIN TEMPERATURE PREVIOUSLY DEFINED. HOWEVER, WE
C MAY STILL USE ALL CHANNELS (1550 LCCP) AND WE MAY STILL DO
C PTE DOWN TO CLIPFACE (1463 LCCP)
C
C IFLC=99
IF (CTYP.EC.*C) GC TO 660
IF (IGES.EC.1.AND.KCSFC.ME.0) GC TO 660
IF (IUSE(12).*E.1) GC TO 660
C CHECK REFLECTED SUNLIGHT. PUT FOR HIGH WINDOW VALUES GIVE
C THE NOD TO CLEAR
C
C WDTF=VCAT(12)-TS
IF (WDTF.GE.1) GC TO 660
IF (VCAT(8).*F.272.AND.WDTF.GT.20) GC TO 660
C CHECK SURFACE AIR AGAINST SKIN ESTIMATE. LOCSEN CHECK
C FOR WARM SKIN (PROBABLY CLEAR) OR FOR VERY COLD SURFACE
C AIR (PROBABLY INVERSION CONDITION) CR FOR FCCR SURFACE AIR EST.
C \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC \nC
CALL VASTAL(T, S, TCTC, V2EN, TAL, ISAT, IICV)

1080 RHF = (1STEF) * VECTAL (TAL, E, ES, IICV, NLS)
TGR = (FLYV-RHFC(2))/(RRHFC(3)-RRHFC(2))
TCP = AMAX1(TCR, +.4)
TGR = AMIN1(TGR, +.2)
RH = RRHFC(2) * ((RHFC(3)/RHFC(2)) + TGR)
CONTINUE

PEO = RRHFC(2)
IF (IGEC.EQ.1. AND. FH.GT.1.2) GC TC 1100
RH = AMIN1(RH, +.1)
RH = AMAX1(RH, +.1)
IRH = IROUND(RH, +100)
IHUM(KC) = IROUND(RH, 1000)
IF (LCRT.EQ.0) GO TO 1100
CALL ENKCEC('"TS."CHAN RWV 100 P50 100 TGP 100 CH_ 100")
LELF
CALL VTCG(LELF)
IRWV = IROUND(RWV, 100)
IKEO = IREAD(ES50, 100)
ITGR = IROUND(TERC, 100)
CALL ENKCEC('"TH1, TH1, TH1"), LEUF, IVC, IRWV, IR50, ITGR, IFR
CALL VTCG(LELF)

1100 CONTINUE

IF (IHUM(3).EQ.WMSG) GC TC 2380
IF (IHUM(1).EQ.WMSG) IHUM(1) = 10000.*WSAT/WSAT(PSTA,TSTA)
IF (IHUM(2).EQ.WMSG) IHUM(2) = (IHUM(1)+IHUM(3))/2
PSSH = IHUM(1) + 0001
C APPORTION THE RELATIVE HUMIDITY TO LAYERS
C IF (POSTC.EQ.0) GC TC 1160
RHSV = WSTA/WSAT(PSTA, TSTA)
IF (RSTA-LT.ESCC) IHUM(1) = IROUND(10000.*RHS)

1160 CONTINUE
CG 1180 I=1,3

1180 IHUM(I) = 0.0001 FLCAT(IHUM(I))
IF (FORCE.EQ.0) GC TC 1160
C IF (HUM(1).GT.CT.0.95) GC TC 2200
C *** I IS LETTING HIGH RH GO HEEAH
C RESTORE SATURATED MIXING RATIO PROFILE
C AND ESTABLISH MIXING RATIO PROFILE
CG 1300 I=26, AL

S(I) = WSAT(F(I), T(I))
IF (I.LT.LT.8CC) GC TC 1260
DRCF = (PSSH-IHUM(I))/(-800.*)
RH = IHUM(1) + DRCF .*(F(I)-8CC)
W(I) = RH**S(I)
IF (F(I).GT.F(IS)) W(I) = PSSH*SC(I)
CG TO 1300

1260 IF (F(I).LT.8CC) GC TC 1260
DRCF = (IHUM(1)-IHUM(2))/(-600.*)
RH = IHUM(2) + DRCF .*(F(I)-600.)
W(I) = RH**S(I)
CG TO 1300
C \[ 1280 \] CRDF=(HUM(2)-HUM(3))/(400.-300.)
C \[ 1282 \] RHF=HUM(2)+CRDF*(P(I)-300.)
C \[ 1283 \] W(I)=RHF*S(T)
C \[ 1284 \] 1300 CONTINUE
C \[ 1285 \] IF(KSUG.EQ.0) GO TO 1320
C \[ 1286 \] CALL ENKCEE('(',T11,')',-15,F7.3,')',LBUF,W(26))
C \[ 1287 \] 1320 CONTINUE
C \[ 1288 \] DC 1322 K=1,4
C \[ 1289 \] KC=K+6
C \[ 1290 \] CALL VASTAL(T,W,TCTC,VZEN,TAUW(1,K),ISAT,KC)
C \[ 1291 \] 1322 CONTINUE
C \[ 1292 \] CALL PRECK(F,W,U,NL)
C \[ 1293 \] C BEGIN RETRIEVAL LCCFS
C \[ 1294 \] 1325 ITLIM=3
C \[ 1295 \] IF(ICES.EQ.1) ITLIM=10
C \[ 1296 \] ITLIM=IKWP(*ITR*,1,ITLIM)
C \[ 1297 \] ITER=0
C \[ 1298 \] IPASS=0
C \[ 1299 \] ICC=0
C \[ 1300 \] C ICK IS SET TO -1 FOR FAILED ENHANCEMENT, +1 FOR SUCCESS IN VTRET
C \[ 1301 \] C INIT=0
C \[ 1302 \] C INIT IS SET TO FIND LOWEST LEVEL JUST ONCE
C \[ 1303 \] C ...
C \[ 1304 \] C BEGIN IMPLICIT ITERATION LOOP
C \[ 1305 \] C IPASS=0 CALC TAU; EC T ITERATION; ENHANCE H2O..CONVERGENCE
C \[ 1306 \] C TEST IS 161520
C \[ 1307 \] C =1 ENHANCE TEMP; LAYER 1
C \[ 1308 \] C =2 LAYER 2
C \[ 1309 \] C =3 LAYER 3
C \[ 1310 \] C =4 REPEAT T ITERATION IF ENHANCE FAILED; REPEAT H2O; EXIT
C \[ 1311 \] C HEAVE ANY CLCFLY/CLIMATOLOGY SOUNDING
C \[ 1312 \] C IF(ICES.EQ.1.AND.IFLG.NE.0)GO TO 2200
C \[ 1313 \] C CHECK ON CALIBRATION AGAIN
C \[ 1314 \] KC2=6
C \[ 1315 \] KC1=1
C \[ 1316 \] IA=IS
C \[ 1317 \] DC 1380 I=1,6
C \[ 1318 \] KUSE(I)=1
C \[ 1319 \] J=7-I
C \[ 1320 \] IF(KC1.EQ.I.AND.IUSE(I).EQ.0) KC1=I+1
C \[ 1321 \] IF(KC2.EQ.J.AND.IUSE(J).EQ.0) KC2=J-1
C \[ 1322 \] 1380 CONTINUE
C \[ 1323 \] IF(ICES.NE.1)GO TO 1385
C \[ 1324 \] IF(IFLG.NE.0)GO TO 2200
C \[ 1325 \] GO TO 2190
C \[ 1326 \] 1385 IF(KC1.EQ.2.AND.KC2.LE.3) GO TO 2200
C \[ 1327 \] DC 1520 KC=KC1,KC2
C \[ 1328 \] ICV=ICT(KC)
C \[ 1329 \] C ...
C \[ 1330 \] C CALCULATE TAU UNDER THE FOLLOWING CONDITIONS:
INITIAL ENTRY

ENHANCEMENT OR 2ND ITERATIVE PASS (ENHANCEMENT FAILED)
BUT ONLY FOR CHAN 4 AND 5 UNLESS GUESS IS CLIMATOLOGY

IF (ITER.GT.0.AND.IPASS.NE.1) GO TO 1420
DELT(KC) = 0.

IF (IPASS.EQ.1.AND.KC.NE.4.AND.KC.NE.5.AND.IGES.NE.1) GO TO 1420
CALL VASTALT(KW,TOTC,VZER,TAU,ISAT,ICV)
DC 1400 IK = 1, 40
WG(KIC,KC) = 0.

1400 TAU(KC) = TAL(IK)
GO TO 1460

1420 DO 1440 IK = 1, 40
1440 TAU(IK) = TALS(KIC,KC)
CONTINUE

IF (INIT.GT.0) GO TO 1470
INIT = 1

IN THE FOLLOWING WE DETERMINE THE LOWEST LEVEL TO WHICH WE
WILL DO RADIATIVE TRANSFER.

TA = TS
IF (IFLG.EQ.1) GO TO 1465
TA = VDAT(R)

IF (IGES.EQ.1.AND.NCSC.NE.0) GO TO 1465

LOOK FOR NEW CLIMUE LEVEL FOR SURFACE TERM

DO 1463 J = 1, 20
I = NLP1 - J
DT = T(I) - TA
IF (CT.GE.0.) GO TO 1463
IA = I + 1
IA = MING (IA, IS)

GO TO 1465

CONTINUE

IF (KRUC.NE.0) CALL ENKCDEN(*, "SKIN LEVEL AND TEMP", IS, F7.27) LREF, IA, TA

1470 CALL VAS GurSTAU(TAU, T, TA, CRAC, E, TBB, CBED, ICV, IA)
E = E1(1)
T1 = TAU(1)
DO 1480 L = 2, IS
B2 = E(L)
T2 = TAU(L)
WCL* = 5.0*(B1*B2)*(T1 - T2)
WCL(KIC,KC) = WCL/CBED
E1 = R2
T1 = T2

SAVE PREVIOUS CONVERGENCE STATE
IF (KUSE(KIC).OR.C) GO TO 1520
IF (KUSE(KIC).OR.C) GO TO 1520
DELS(KIC) = CELT(KIC)
DELT(KIC) = VCAT(ICV) - TBB
1520 CONTINUE
1530 IF (ITCRED GT.6) GC TO 1550
1540 IF (IUCS(E).EQ.0) GC TO 1540
1550 IF (IFLAG.NE.0) GC TO 1540
1560 C WATCH OUT FOR SHAKY CHANNEL 6 WHICH WE DON'T UNDERSTAND
1570 ABE=ABS(DELT(5))
1580 AB6=ABS(DELT(6))
1590 IF (ABE.LT.ABE) GC TO 1530
1600 AB4=ABS(DELT(4))
1610 IF (ABE.LT.AB4) GC TO 1530
1620 ICSE(6)=0
1630 C TEST CHANNEL 5 VS 6 RESIDUAL AND ASSUME THAT A LARGE
1640 C DISCREPANCY INDICATES CLOUDY PROBLEM (HIGH RH1 AS WELL)
1650 1530 CCK=DELT(5)-DELT(6)
1660 ICCK=100. CCK
1670 IF (KREU.NE.0) CALL SDEST("CHAN5-CHAN6= *,ICCK)
1680 RHCK=MAX1(HUM(1),HUM(2))
1690 IF (ABE(CK).GT.2.E.KANK.RHCK.GT.0.95) IFLAG=1
1700 1540 CONTINUE
1710 C FOR HIGH CLOUD COVER WE THROW OUT SHORT WAVE CHANNEL
1720 IF (PSA.LE.850.) KLSE(6)=0
1730 C SET ASICE ONE CHANNEL FOR TESTING SUCCESS OF RETRIEVAL(81960)
1740 C ITCH=5
1750 IF (IFLGC.NE.0) GC TO 1520
1760 IF (CTYF.EQ.5) GC TO 1580
1770 IF (IGFS.EQ.1.AND.KCSCF.NE.0) GC TO 1560
1780 IF (IFLGC.EQ.0) GC TO 1560
1790 C ENTER FOR SUSPECTED CLOUDY SCANNING WITH REASONABLY GOOD GUESS
1800 C CHECK CBS-CALC FOR INCREASINGLY CLOUDY CHANNELS AND DELETE THOSE
1810 C COLDER THAN GUESS
1820 DC 1560 I=KC1,KC2
1830 JU=KC2+1-I
1840 J=JU
1850 C REVERSE ORDER BECAUSE CHAN 5 IS MORE TRANSPARENT
1860 IF (JU.EQ.6) =5
1870 IF (JU.EQ.5) =6
1880 DELTT=DELT(C)
1890 IF (DELTT.EQ.-C.25) GC TO 1550
1900 IF (KREU.NE.0) CALL SDEST(" REMOVING CLOUDY CHANNEL ")
1910 KLSE(J)=0
1920 C SET RETRIEVAL TYPE TO OVERCAST AT THIS POINT
1930 IYF=23
1940 CC TO 1560
1950 C IF 8 PASSES SKIP CLT BUT CHECK 4 EVEN IF 6 PASSES
1960 1550 IF (JU.EQ.6) GC TO 1580
1970 1560 CONTINUE
1980 1580 CONTINUE
1990 IF (KUSE(4).NE.0) GC TO 2200
2000 IF (KREU.EQ.0) GC TO 1660
CALL EKKCDE(T11 *, T = "15F7.1", LEUF, T26)
CALL EKKCDE(T11 *, T = "15F7.3", LEUF, W26)
CALL EKKCDE(T11 *, T = "EF8.2", LEUF, DELT)

1660 CONTINUE C
IF RECALCULATION ON SECOND PASS SHOWS LARGE RESIDUAL ASSUME
THAT SOMETHING HAS GONE WRONG
C
DC 1661 KC=KC1*KC2
IF (IUSE(KC) .EQ. 0) GC TO 1661
IF (KUSE(KC) .EQ. 0) GC TO 1661
IF (FORCE*CT.C) GC TO 1661
IF (IPASS.ME.C.ANC.ABS(CELT(KC)).GT.2.5) GC TO 2200

1661 CONTINUE
IF (IPASS.ME.C) GC TO 1662
IF (ICK.GT.0) GC TO 1640
C ABOVE JUMPS CTR WHEN ENHANCEMENT OF T COMPLETED SUCCESSFULLY
C OTHERWISE CONTINUE THE ITERATIVE WITH NEW TAU UNLESS CONVERGED
C GC TO 1670

1662 IF (IPASS..EQ.0) GC TO 1670
C CHECK CONVERGENCE OF ALL CHANNELS WITHIN NOISE
DC 1665 I=KC1*KC2
DTABS=ABS(CELT(I))
IF(DTABS.GT.EX(I)) GC TO 1668

1665 CONTINUE
C GC TO 1951

1668 CONTINUE
IF (IFLGP.ME.0) GC TO 1940
ICK=-1
IF (IENH.ME.1)
CALL VTRRET(KC1, KC2, DELT, KUSE, WJ, IS, IPASS, EX, ICK)
IF (ICK.LT.0) GC TO 1882
GC TO 1730

1670 CONTINUE
C
C APPLY CORRECTION TO TEMP PROFILE
DO 1720 I=2,IS
FNUF=0.
FDEF=0.
DC 1700 J=KC1*KC2
IF(J.EQ.ITCH) GC TO 1760
IF(IUSE(J) .EQ. 0) GC TO 1700
WCHT=WG(I,J)/EX(J)
IF(KUSE(J) .EQ. 0) GC TO 1680
FNUF=FNUF+CELT(J)+WCHT

1680 FCE=EXP(WCHT+FCE)

1700 CONTINUE
IF(ABS(FCEA) .LT. 1.E-8) GC TO 1720

1720 CONTINUE
C CHECK FOR CESS T(P) BLENDING
IF (PREF.GE.P(IS)) GC TO 1720
WPROF=((1-1)*PREF)/(P(IS)-PREF))
T(I)=WPROF TSAVE(I)+(1-WPROF)*T(I)
1730 CONTINUE

ADJUST LOW LEVEL T(P) USING SURFACE AIR TEMP

IF (DSFC.LE.0) GC TO 1840

DCS=TSTA-T(I)

IF (IS.LE.57) GC TO 1800

C IF SURF AT CR ABOVE 650 EXIT, OTHERWISE BLEND LOWER LEVELS

C WITH SURFACE AIR TEMP RETAINING SHAPE OF PROFILE

C DTDF=DCS/(F(IS)-P(37))

DC 1760 I=37,15

1780 T(I)=T(I)+CTCF*(F(I)-P(37))

1800 DC 1820 I=IS,40

1820 T(I)=TSTA

1840 CONTINUE

DC 1860 I=26,40

1880 IF (S(I).LT.C.02) GC TO 1880

SNEW=SAT(F(I),T(I))

1900 IF (SNEW.LT.C.02) SNEW=C.02

W(I)=W(I)*SNEW/S(I)

S(I)=SNEW

1960 TC(I)=CEWFT(F(I),T(I),W(I))

1980 CONTINUE

IF (IPASS.LE.0.OR.IPASS.GT.3) GC TO 1885

1982 IPASS=IPASS+1

IF (ICK.GE.3) GC TO 1390

IF (IPASS.LE.3) GC TO 1668

C ABOVE AVOIDS REPEATING RAD TRAN WHEN ENHANCEMENT FAILED

C FOR FAILED ENHANCEMENT APPLY NORMAL ITERATIVE CORRECTION

ICK=0

ITLIM=ITLIM+3

C RAISE ITERATION LIMIT WHEN WE GO TO SECOND PASS OF ITERATIVE

G C TO 1670

1885 CONTINUE

C CHECK CONVERGENCE

ICK=0

D2=0.

SUM=0.

DC 1920 KC=KC1,KC2

IF (KUSE(KC).LE.0) GC TO 1920

1970 IF (ICT(KC).LT.ITCH) GC TO 1900

KSAV=KC

1980 GO TO 1920

1990 D1=ABS(KF(1)-CELS(KC))

1992 DC 2000 D2=D2+1.

1995 SUM=SUM+D1+D1

1998 1920 CONTINUE

1999 SUM=SQRT(SUM/D2)

ITER=ITER+1

IF (SUM.LT.C.0.25) GC TO 1540

1999 IF (KPLC.GE.0.5) CALL SDECST(*,END ITER = *,ITER)

2023 IF (ITER.GE.ITLIM) GC TO 1840

2030 GO TO 1350
1940 CONTINUE

IF (IENF, NE, 0) GC TO 1949

C

BEGIN MOISTURE ENHANCEMENT

REDEFINE CLOUD LEVEL

LS=NL

CC 1942 J=1,20

I=NLPS1-J

DT=T(I)-TS

IF(DT.GT.0.) GC TO 1942

LS=I-1

LS=MINC(LS, NL)

CC TO 1943

1942 CONTINUE

1943 CONTINUE

PLS=MINC(.IS, LS)

DO 1944 K=1,4

KC=K+6

CALL VASTAL(T, W, TSTC, VZEN, TAUW(1), KSAT, KC)

1944 CONTINUE

CALL VWRET (TS, TAUW, U, NL, NLS, TSP, USP)

1949 CALL PRECW(F, T, NL)

C

DO 1950 I=26,40

1950 TC(I)=DEWFT(F(I), T(I)+W(I))

IF(IPASS.NE.0) GO TO 1951

IF(IPASS.EQ.0) IPASS=1

C

RETURN TO RECALCULATE TRANSMITTANCE AND ATTEMPT ENHANCEMENT

GO TO 1360

1951 CONTINUE

TCT=(37)+TC(37)-2*T(31)

IF (FORCE.NE.0) GC TO 1960

CTST=2.5

IF (ITCH.EQ.E) CTST=2.0

660 IF (TS.GT.305) CTST=5.0

661 IF (KUSE(ITCH).EQ.0) GC TO 1960

662 IF (VDA1(ITCH).EQ.WISG) GC TO 1960

663 TEST=ABS(CELT(KSAY))

664 IF (TEST.GT.70) GC TO 2200

1960 CONTINUE

CALL HTX(IS, IS)

C

SET UP FOR STABILITY CALCULATION

668 NB=IS

669 IF (PSL1.LT.F(NB)) NB=NE-1

670 GC 2010 I=2+15

671 PST(I)=F(NB)

672 TST(I)=T(NB)

673 TCST(I)=TC(NB)

674 2010 NE=NB-1

675 PST(I)=FSTA

676 TST(I)=.81 ITSFC
TEST(1) = .01 ID$FC
IF (ID$HC(1)) CALL ENKCEE('"STAB, PRESS",10F8.2)', LBUF,TST(1)
CALL SUDAIL(*15,PST,TST,TDT,TST,DIR,SPD,STABIL)
LIFT=100, STABIL(8)
NC 2020 T=1, AL
J=NLPL-1
U(1)=S(I)
C ABOVE S(I) GCEES WITH NEW HTX CALL
2020 CONTINUE
DELZ1=L(31)-L(37)
DELZ2=U(24)-L(37)
IF (KBU.EQ.0) GC TC 2060
CALL ENKCEE('"T11,CELZ1 =",F10.2,"X,CELZ2 =",F10.2)"
LEUF,CELZ1,CELZ2)
2060 CONTINUE
IF (KBU.EQ.0) GC TC 2180
CALL ENKCEE('"T6, PRESSURE ",15F7.0)",LEUF,T(26))
CALL ENKCEE('"T6, PROFILE ",15F7.1)",LEUF,T(26))
CALL ENKCEE('"T6, DEW POINT ",15F7.1)",LEUF,T(26))
CALL ENKCEE('"T6, HEIGHT ",15F7.0)",LEUF,L(26))
CALL ENKCEE('"T6, MIX RATIO ",15F7.3)",LEUF,L(26))
CALL ENKCEE('"T11, TCT =",F7.2)",LEUF,T(26))
2180 CONTINUE
C COMPUTE AN AVERAGED TOTALS...
C0=.3337, CELZ2=8457*CELZ1
C1=.14044, CELZ1=.015, CELZ2
T500=C0+C1:ALCG(500)
T850=CC+C1:ALCG(850)
P850=.660:FLCAT(I-U(1))
P500=.50.
WE50=R500, WSAT(P850,T850)
D500=DEWFT(FF50,T500)
TCTLS=1850+D500-2,.T500
T850=IRCLAC(T850,100)
IC850=IRCLAC(C850,100)
IT500=IRCLAC(T500,100)
ITCTLS=IRCLAC(TCTLS,100)
IDZ1=1PCUNC(CELZ1,100)
KDZ2=1PCUNC(CELZ2)
IDZ2=IRCLAC(CELZ2)
IF (LCLRT.EQ.0) GC TC 2190
CALL VTG(LLEF)
CALL ENKCEE('"(618)",LBUF,T850,IC850,IT500,TCTLS,KDZ1)
CALL VTG(LLEF)
2190 NCNO=0
2200 CONTINUE
C PREPARE CUTFLT BUFFER
IPETD(3)=7777
IF (NCNO.NE.0) GC TC 2360
VLSRET

726 C IRETD(3)=C
727 C ABOVE IS USER MODD FLAG TO CONTROL PACKING
728 C CALCULATE FINAL TEB
729 C THESE CALCULATIONS ARE DEACTIVATED TO SAVE TIME...
730 DC 2205 K=KC1*KC2
731 IF (KUSE(X) EQ 0) GO TO 2205
732 IF (ICE.EG0) GO TO 2205
733 CALL VASTAL(T, W, TCTC, VZEN, TAU, ISAT, K)
734 CALL VASTEPE(TAU, T, TS, CPAD, E, TEB, CEDE, K, IA)
735 IRETD(21+K)=100.* TEB+0.5
736 2205 CONTINUE
737 DC 2210 K=7.1C
738 C F=6
739 IF (ICE.EG0) GO TO 2210
740 CALL VASTAL(T, W, TCTC, VZEN, TAUW(1, J), ISAT, K)
741 CALL VASRTE(TAUW(1, J), T, TS, CRAD, E, TEB, CBDB, K, IA)
742 IRETD(21+K)=100.* TEB+0.5
743 2210 CONTINUE
744 CALL PRECK(F, K, L, IS)
745 IRETD(35)=L(IS)-1000+0.5
746 IRETD(36)=1000+TS
747 IF (ICE.EG0) GO TO 2218
748 IF(PSTA=GE.EG0) IRETD(41)=TOT*100.
749 IRETD(42)=11TCLS
750 IRETD(43)=LIFT
751 DC 2211 K=1,3
752 IF(IHUM(K).EQ.100000) IHUM(K)=MISG
753 IF(IHUM(K).EQ.200) IHUM(K)=MISG
754 2215 IRETD(43+K)=IHUM(K)
755 2218 CONTINUE
756 IRETD(50)=ITYF
757 LS=NLP1-IS
758 IF (LS*LT.2) LS=2
759 DC 2220 K=2,20
760 L=L*O(K)
761 M=(K-1)*9
762 N=NLP1-L
763 IRETD(57+M)=S(L)
764 C ABOVE CHANGED FROM A TO C L TO CONFORM WITH HTX
765 IRETD(51+M)=LCHR(K)
766 IRETD(52+M)=F(L)-10.
767 IF (P(L).GT.PSTA) GO TO 2220
768 C STORE ONLY HEIGHTS BELOW SURFACE
769 IRETD(53+M)=T(L)-100.
770 IRETD(58+M)=TCS(K)*100.
771 IF (K.GT.11) GO TO 2220
772 IRETD(54+M)=T(L)+100.
773 IRETD(55+M)=TCS(K)*100.
774 2220 CONTINUE
775 C ADD SURFACE VALUES
776 IRETD(51)=LCHR(1)
777 IRETD(52)=PSTA 10
781  IRET(52)=ITSFIC
782  IRET(54)=ICSCFC
783  IRET(57)=IELEV
784  IRET(58)=TCS(1):1CC.
785  IRET(59)=ECS(1):1CC.
786  C       MD CUTFLG = DIFFER COMPLETE
787  ICET=4,RET+1
788  IF (IFLT.EQ.0)GC TC 236C
789  C       REVERT TO -M-E CONVENTION
790  SLON=-VLCN
791  CALL SETAR(FTIME,FLIN,FELE,VLAT,SLON,2,INAV,RETAIN,RETDOG,0.)
792  TL=FLIN*0.5
793  IE=FELE+0.5
794  CALL SATTV(IFRM,IL,IE,IRAS,IFIC+SAT,JDAY,UTIME)
795  JDAY=JSAT 1CCCCCE+JCAY
796  KOLCR=2
797  IF (ITYP.EQ.22)KOLCR=3
798  IF (IFLG.NE.0) KOLCR=1
799  ITOLLS=(ITOLLS+5)/100
800  CALL VASIFIC(IRAS+MAC2,IFIC+MAG2+ITOLLS,MAG,1,KOLCR)
801  236C  CONTINUE
802  CALL VRTIC(IRET,NN,1)
803  238C  CONTINUE
804  240C  CONTINUE
805  250C  CALL SELECT(* LAST RETRIEVAL NO. PROCESSED WAS *,NN)
806  RETURN
807  END
C CRT DISPLAY CF VAS-RETRIEVAL GUESS AT CURSOR POSITION
C ? DISPLAY VAS GUESS PROFILE AT CURSOR LCC OR AT SPECIFIED LAT/LON (CM)
C ? KEYIN:: GPVA <CSS> <SFC> <LAT> <LON> <MONG>
C ? POSITIONAL PARAMETERS
C ? CSS - 'C' 'F' FOR CLIM OR FRCF..DEFAULT TO MONG
C ? SFC - NC SURFACE ANALYSES IF SET TO NON ZERO INTEGER
C ? LAT - LATITUDE=100
C ? LON - LONGITUDE=100 (POS W, NEG E)
C ? MENG - NC FILE FOR GUESS..DEFAULT TO 'VASTEXT' ENTRY
C SSEC/MCIRAS USERS MANUAL - CHAP12
C DIMENSION KCLT(10),LF(15),VFAT(13)
C DIMENSION IFER(112)
C COMMON/ATMCS/F(40),T(40),W(40)
C COMMON/CLTG/CS(15),CS(6)
C COMMON/GUESS/TGES(15),TGES(6)
C COMMON/NV/VALE,VLON,VEPA,VEPA,AS,IPIC,ITIM,UTIM,JDAY
C COMMON/SLRF/I211,ITSFC,IDSFC,IPSTA,IEL,LSTA
C CHARACTER 12 CPF,CGES
C CHARACTER E MFIL
C DATA MFILE /*VASTEXT */LUN/20,LEN/100/
C DATA MISS/780808080/
C DATA IF/I1CC,50,7CC,50,400,300,250,200,150,100,70,50,30,20,10/
C DATA LF/4,7C,3E,31,28,26/
C CGES=CPF(1,*,*)
C NCFLC=IPPA(2,0)
C KBUG=IKWF(*EKG*,1,0)
C ICLIM=0
C IIF(CGES:*E'C') ICLIM=1
C IIF(CGES:*E'F') ICLIM=2
C VLAT=01,FLCAT(IPPA(2,0))
C VLON=01,FLCAT(IPPA(4,0))
C IIF (LUC(81),NE,0) IEC TO 111
C CALL TSNIC(1,1,1,1,1,1,1,12)
C TO 112
C 111 ITERM=IKWF(*TERM*,1,0)
C IIF(ITEM,EG,0) ITERM=LUC(-20)
C READ VASTEXT FOR MT INFO
C CALL DCOPEN (MFILE,LLN,LEN)
C CALL DREAD(LLN,ITEM,IECC)
C CALL DCLOSE(LLN)
C 112 MENG=IPPA(5,0)
C IIF(MENG,EG,0) MENG=IECC(18)
C IIF (VLAT,NE,0) GO TO 3
IFRM=LLC(-1)
IRAS=LLC(-11)
IPIC=LLC(-12)
CALL TVSAT(IFRM,IRAS,IPIC,IL,IE,IS,JDAY,UTIME)
C PUT THE SS BACK INTO SSYDDCC
JDAY=US 100000+JDAY
IF (LLC(81),EG,(!)GC TO 115
VDAT(1)=1.
C GET VAS NAVIGATION IF REQUIRED
CALL VASCAT(IL,IE,VCAT)
3 LAT=IROND(VLAT 100.)
C HRTopo NEEDS LCN+ EAST
LCN=IROND(-VLON-100.)
GO TO 120
115 CONTINUE
CALL TSNIC(1,1,1,1,1,1,1,IHDR)
CALL TSNIC(C,1,IL,IE,1,1,LAT)
CALL TSNIC(C,2,IL,IE,1,1,LCN)
VLAT=.01 LAT
C COMMON /NAV/ NEEDS VLON + WEST
VLON=-.01 LCN
120 CONTINUE
CALL HRTFSC(LAT,LCN,IEL,ICH)
IF (ICIM,EG,(!)CALL SDEST(" CLIMATOCICAL GUESS",0)
CALL GESFSC(ICIM,MCSFC,MENC)
CALL SDEST(" LAT LCN ELEV PSTA Z1000 TSFC
SFC",0)
KOUT(1)=7
KOUT(2)=LAT
KOUT(3)=LCN
KOUT(4)=IEL
KOUT(5)=IPSTA
KOUT(6)=IZ1C
KOUT(7)=ITSFC
KOUT(8)=ICSF
CALL CUTFINT(KCUT)
CALL SDEST(" LEVEL PRESS TEMP DEWP LWIX",0)
DC 180 I=1,6
KOUT(1)=5
KOUT(2)=I
KOUT(3)=IF(I)
KOUT(4)=IRCLAD(TGES(I)+100.)
KOUT(5)=IRCLAD(CCES(I)+100.)
J=LP(I)
KOUT(6)=IRCLAD(J(J)*100.)
CALL CUTFINT(KCUT)
180 CONTINUE
DC 200 I=7,15
KOUT(1)=3
KOUT(2)=I
KOUT(3)=IF(I)
KOUT(4)=IRCLAD(TGES(I)+100.)
CALL OUTIT(KCUT)

200 CONTINUE

RETURN

END
C    EDIT SURFACE REPORT AT CURSOR LOCATION
C    KEYWORDS:
C    "Z1000" 1000M PRE HEIGHT VALUE
C    "TSL" TEMPERATURE AT SEA LEVEL
C    "DEW" DEWPOINT DEPRESSION
C    "TSFC" SURFACE TEMP TO BE CONVERTED TO TSL
C    IF NO PFT. EXISTS IN CURSOR, ONE IS ADDED; OTHERWISE VALUES
C    OF KEYWORDS ARE SUBSTITUTED. VALUE OF 1 ENTERS "MISS" (EDITS).
C    N.B. TSL AND DD MUST BE >= 100, Z100 IS W
C
SSEC/MCICAS USERS MANUAL - CHAP12
DIMENSION MPF(64), IOUT(25), IOUT(25)
DIMENSION ILIST(25), ISCL(25), IUN(25), ILC(25)
DIMENSION KCU(10)
INTEGER 4 CEUF(20)
CHARACTER*8 FILE
REAL*8 DLIT
COMMON /DCDC/ICDC(112)
COMMON /NAV/FLAT, FLON, ZENLCC, ZEN, IL, IE, IRAS, IFIC, IHMS, JT, JC
COMMON /IERT/IYMC, JHMS, NROW, NSAT
COMMON /ANKL/NOAN, LTOP
COMMON /CREIT/NCDE
COMMON /TIGHT/ITCL
COMMON /ENTRY/INIT
COMMON/ORIENT/YCCC, XCCC
COMMON/DBLC/KBLC
DATA MPF, IL, LEN/*VASTEXT */20, 100/
DATA ICUT /25 ZFO080C80/, IOUT/25 ZFO080C80/
DATA MISS/ZFO080C80/, ILC/56, ILC/1, RDAT/0/
IWRM=LUC(-5)
IFRMS=LUC(-1)
IPASM=LUC(-11)
IPICM=LUC(-12)
CALL TVSAT(IFRMS, IRAS, IFIC, IL, IE, JS, JD, JT)
C SAVE LOCATION FOR POSSIBLE ADDITION
ALIA=FLCAT(IL)
AEEL=FLCAT(IE)
IF (LUC(81)=.EQ.0) GO TO 110
C READ VASTEXT CONTEXT FILE
ITERM=LUC(-20)
CALL DCFEN(CLIT(MFILE), LEN)
CALL DREAD(LIA, ITERM, ICDC)
CALL DCLSE(LIA)
C INITIALIZE NAVIGATION
JC=JS*100000+JD
CALL NVINIT(RETAIR, BETDCT, INAV, FTIME)
CC TO 120
110 CALL TSIC(1,1,1,1,1,1,1,1,IEUC)
   INIT=1
   NCODE=IDCC(7)
C SET UP AREA FOR IMAGE SPACE
   CALL TVSAT(IFRM,005,335,LTCP,LELE,ISS,IC,IT)
   CALL TVSAT(IFRM,455,335,LECT,LELE,ISS,IC,IT)
   IF (LTCP.LT.1) LTCP=1
   IF (LECT.LT.1) LECT=1
   NROWS=LECT-LTCP+1
   NPTS=NROWS*ACCLS
120 CONTINUE
   MDNO=IDCC(36)
   MDR=IDCC(37)
2 IF (MDCPEN(MDCG,2),NE.0) GO TO 900
C READ RCW HEADER RECORD
   IOK=MDGET(MDCG,MDR,F,ICUT)
   IF (IOK.LT.0) GO TO 502
   MREC=ICUT(3)
C READ TEST RECORD
   N=0
3 N=N+1
   IF (M.GT.10) GO TO 504
   IOK=MDGET(MDCG,MDR,N,ICUT)
   IF (IOK.LT.0) GO TO 3
   KBUC=IKWP(*ELG*,1,0)
   IF (KBUC.NE.0) CALL ICEST(* OPERATING WITH RCW=*,MDR)
60 C READ IN KEYS
   NKEYS=MKEYS(MDCG,-1,LIST,ISCL,IUN,LCCS)
C FIND DATA ADDRESSES IN KEYLIST
   DO 11 N=1,NKEYS
54 IF (LIST(N).EQ.LIT(*LAT *))NLAT=N
55 IF (LIST(N).EQ.LIT(*LON *))NLON=N
56 IF (LIST(N).EQ.LIT(*MS ))NTIM=N
57 IF (LIST(N).EQ.LIT(*Z100 *))NZ=N
58 IF (LIST(N).EQ.LIT(*TSL *))NK=N
59 IF (LIST(N).EQ.LIT(*CD *))ND=N
11 CONTINUE
91 KOUT(1)=5
92 KOUT(2)=NLAT
93 KOUT(3)=NLON
94 KOUT(4)=NZ
95 KOUT(5)=NK
96 KOUT(6)=ND
97 IF (KBUC.NE.0) CALL CUTINT(KOUT)
98 IZ100=IKWP(*Z100 *,1,0)
99 ITMSL=IKWP(*TSL*,1,0)
100 C ADD OPTION TO KEY IS SURFACE TEMP
101 ITSFC=IKWP(*TSFC*,1,0)
102 ICD=IKWP(*EC*,1,0)
103 IF (IZ100.EQ.1) IZ100=MISS
104 IF (ITMSL.EQ.1) ITMSL=MISS
IF (ICD.EQ.1) ICCC=MISS
ITEM=0
IF (IZ1CC.EQ.0) ITEM=IZ100
IF (ITMSL.EQ.0) ITEM=ITMSL
IF (ICD.EQ.0) ITEM=ICD
IF (ITSFC.EQ.0) ITEM=ITSFC
IF (ITEM.EQ.0) CC TC 80
IF (ITEV.EQ.MISS) ITEM=999
IF (KIRG.EQ.0) CALL SDEST (* INPUT CATLM = *ITEM*)
CALL GETFRM(IFRM,IF)
MAG=F(10)
IF (MAG.LT.1C) MAG=10
CALL INITPL(IFRM,0)
IF (LUC(L1).EQ.0) CC TO 6C

C FOR VAS GET CURSOR DIMENSIONS IN LAT/LCN
IRINC=LUC(-5)
IFINC=LUC(-10)
IRS=LUC(-11)
IPC=LUC(-12)
IPAS=IRS-IRINC/2
IFIC=IPC-IFINC/2
CALL TVSAT(IFRM,IRAS,IFIC,IL,IE,JS,JEC,JT)
KOUT(2)=IL
KOUT(3)=IE
FLIN=FLOAT(IL)
FELE=FLOAT(IE)
CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLCN,1,INAV,BETAIN,BETDOT,CO)
C CONVERT TO +W-E CONVENTION
FLCN=-FLCN
KOUT(4)=ILALC(FLAT)
KOUT(5)=ILALC(FLON)
ILAMAX=IRCLAC(FLAT,-10000.)
ILMAMAX=IRCLAC(FLON,-10000.)
IPAS=IRS-IRINC/2
IFIC=IPC+IFINC/2
CALL TVSAT(IFRM,IRAS,IFIC,IL,IE,JS,JEC,JT)
FLIN=FLOAT(IL)
FELE=FLOAT(IE)
CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLCN,1,INAV,BETAIN,BETDOT,CO)
C CONVERT TO +W-E CONVENTION
FLCN=-FLCN
KOUT(6)=ILALC(FLAT)
KOUT(7)=ILALC(FLON)
ILAMIN=IRCLAC(FLAT,-10000.)
ILMAMIN=IRCLAC(FLON,-10000.)
KOUT(1)=6
IF (KIRG.EQ.0) CALL CUTINTKOUT)
C NAV COMPLETE
GO TO 68
C PICK UP TOVS LOCATION
60 CALL TSMIC(0,1,IL,IE,1,1,LAT)
CALL TSMIC (C, 2, IL, IE, 1, 1, LCN)

ALAT = LAT / 100
ALCN = LCN / 100
ILAMIN = LAT 100 - 2000
ILAMAX = ILAMIN + 4000
ILMIN = - LCN 100 - 2000
ILMAX = ILCMIX - 4000
KCUT(1) = 4
KCUT(2) = ILAMIN
KCUT(3) = ILAMAX
KCUT(4) = ILCMIX
KCUT(5) = ILCMAX
IF (KELG = NE = 0) CALL CUTINT (KCUT)

68 DO 70 N = 1, NREC
ICK = MDCT (MENC, MDR, M, ICUT)
KCUT(1) = 3
KCUT(2) = N
KCUT(3) = ICLT (ALAT)
KCUT(4) = ICLT (ALCN)
IF (KELG = NE = 0) CALL CUTINT (KCUT)
C DELETE EVERY ONE WITHIN CURSOR
IF (ICUT(ALAT) LT ILAMIN OR ICUT(ALAT) GT ILAMAX) GO TO 70
IF (ICUT(ALCN) LT ILCMIX OR ICUT(ALCN) GT ILCMAX) GO TO 70
C THIS SOUNDING MUST GO
FLAT = ICUT(ALAT) - 0.0061
FLCN = ICUT(ALCN) - 0.0001
IF (LUCS (1) LT ECG) GO TO 69
C LOCATE RASTER AND PIXEL OF SOUNDING
C REVERT TO -E+ CONVENTION
FLON = FLON
CALL SATEAR (FTIME, FLIN, FELE, FLAT, FLCN, 2, INAV, EEDAT, EEDOT, 0.)
IL = FLIN + C.E
IE = FELE + 0.5
69 CONTINUE
C USE TOVS NAVIGATION ROUTINE
LATS = FLAT 100
LONGS = FLON 100
CALL SRCH (LATS, LONGS, IM, IL, IE, NPTS, NROWS)
IF (KELG = NE = 0) CALL SEAEST (* SRCH GIVES IV = *, IV)
IF (IM = ECG) GO TO 70
690 CALL SATTV (IFRM, IL, IE, IRAS, IPIC, JS, J=JT)
KCUT(1) = 4
KCUT(2) = IL
KCUT(3) = IE
KCUT(4) = IALC (FLAT)
KCUT(5) = IALC (FLCN)
IF (KELG = NE = 0) CALL CUTINT (KCUT)
IF (IZ100 = NE = 0) ICUT(NZ) = IZ100
IF (ITMSL = NE = 0) ICLT (INT) = ITMSL
IF (IDD = NE = 0) ICLT (AC) = ICD
ICK = MDPUT (MENC, MDR, M, ICUT)
IF (ICK .NE. C) GO TO 506
IRAX = IRAC + 4
IFIX = IFIC + 4
JHI = MAG 4/3
JWI = 1
CALL FLTDIC (IRAX, IFIX, ITEM, JHI, JWI, ICOL)
NDAT = 1
CONTINUE
70 CONTINUE
IF (NDAT .NE. C) GC TC 100
IF (ITEM .EQ. 95) GC TC 907
GC TO 8E
80 CALL SEEEST (* NC ACTION INDICATED AT THIS LOCATION * 0)
RETURn
85 IF (LUC (B1) .NE. 0) CALL SATEAR (PTIME, ALN, AELE, ALAT, ALON, 1, IRAX,
BETAI, DETECT, 0.)
C
PREPARE TO ADD VALUE(S)
IF (IZ100 .EQ. 0) IZ100 = MISS
IF (ITMSL .EQ. 0) ITMSL = MISS
IF (IDC .EQ. 0) IDC = MISS
IF (ITSFC .EQ. 0) GC TC 300
C
CHECK REASURABLENESS OF TEMPERATURE
MLAT = ALAT - 100.
MLON = ALON - 100.
CALL HRTFIC (MLAT, MLON, IEL, ICH)
Z = IEL
TSL = -0.01 * FLCAT (ITSFC) * Z ** 0.0065
ITMSL = TSL 100.
300 CONTINUE
300 CONTINUE
C
CHANGE SIGN OF LONGITUDE FOR MD FILE
ALON = ALCA
JOUT (1) = ICLT (1)
JOUT (2) = ICLT (2)
JOUT (3) = ICLT (3) + 1
JOUT (4) = 0
JOUT (NLAT) = ICLNE (ALAT - 10000.)
JOUT (NLON) = ICLNE (ALON - 10000.)
JOUT (N1M) = ICLT (2)
JOUT (MNZ) = IZ100
JOUT (NMT) = ITMSL
JOUT (NC) = IDC
ICK = MDCUT (MC, MEC, JOUT (3), JOUT (4))
IF (ICK .NE. 0) GC TC 906
ICK = MDCUT (MC, MDC, 0, JOUT (4))
IF (ICK .NE. 0) GC TC 906
IRAX = IRAS + MAG
IFIX = IFIC + MAG
JHI = MAG 4/3
JWI = 1
CALL FLTDIC (IRAX, IFIX, ITEM, JHI, JWI, ICOL)
100 CALL ENPLT
IF (IKNP (PACK .NE. 1)) RETURN
CALL ENKCEE (* (K5, 7X, "CCL, CMAX", 4X, "R0E", "K2", 6X) * QPLF * MDNC * MCR)
CALL JSF,(*MCACK*,OBUF(1),OBUF(4),OBUF(7))
RETURN
500 CALL SDEST(* CANNOT OPEN MCFILE NO. *MDNC)
RETURN
502 CALL SDEST(* CANNOT LOCATE ROW NO. *MER)
RETURN
504 CALL SDEST(* TOO MANY ERRORS IN READING ROW NO. *MER)
RETURN
506 CALL SDEST(* TPCLELY WRITING DATA RECORD NO. *H)
RETURN
507 CALL SDEST(* CANNOT LOCATE REPORT TO EDIT *C)
RETURN
END
// EXECUTE MAIN

C ? KEYWORD (DEFAULTS IN PARENTHESES):

C ? VARIOUS TOTAL PRECIPITATION WATER VAPOR RETRIEVAL (V.T.M.WCCLF)
C ? KEYWORD: VTF2 <KEYWORDS>

C ? APA=NUMBER OF DIGITAL AREA FOR IMAGING (0)
C ? SCX=X SIZE IN FCV'S CF SCR BOX (11) >EVEN-LC, ODD-SCF
C ? SFC=1L IE(LINE AND ELEMENT SPACING CF RTVLS (BOX BOX))
C ? BEG=IL IE(FIRST LINE AND ELEMENT TO DEFINE AREA (VATESTX))
C ? Eag=IL LE(LAST LINE AND ELEMENT TO DEFINE AREA (VATESTX))
C ? NTH=FREQUENCY WITH WHICH TO PUT RTVLS IN MD-FILE (2)
C ? ...
C ? GSS=G (GRID, DEFAULT)
C ? SFC=1L NC SURFACE ANALYSIS (0)
C ? TCH=E (TEMP. CHANNEL (6))
C ? PLT=1 (PLOT RESULTS ON GRAPHICS (0))
C ? CRT=1 (FCR CRT OUTPUT (0))
C ? BLC=1 (FCR DEBUG DIAGNOSTICS (0))
C ? TEL=L (TERMINAL NUMBER (LOCAL))
C ? THE FOLLOWING POINTERS MUST BE SET WITH *SFVA*:
C ? LNAT,MDEH,MDDH,N RET ... ALWAYS
C ? ...
C ? NOTE: MDR MUST BE CREATED WITH SCHEMA *VFVW*, NCT VRET*

DIMENSION LEFL(33),NSPIN(13),VDAT(13),VAS(11,11)
DIMENSION IRET(250),IMEST(8),IARET(660),IABUF(165),AFAIL(7)
DIMENSION FARRAY(11,11),LFLAG(11,11),RATIO(11)
INTEGER 2 IARRAY(11,11,60)
CHARACTER 12 CGES, CKWP, CEVE, CLET, CLETG
COMMON/ APT, ICIF (64)
COMMON/ ATRCE,F(40),T(40), W(40)
COMMON/AUTO/ BOX, ILO, ILE, IER, MAG
COMMON/ DANGLE/ VLATS, VLOS
COMMON/ DEUC/ LEUG
COMMON/ DOC/ ICFC (108)
COMMON/ CEV, EV(12), EV(12), EV(12)
COMMON/ GUES/ TGES(15), CGES(6)
COMMON/ LAST/ LASLIN, LALE, LELEV, ICHAR
COMMON/ MODE/ ICT(13), ISFIN(15)
COMMON/ NAV/ VLAT, VLOK, VZEN, SZEI, IL, IE, IRAS, IFC, ITIME, JTIME, JCAT
COMMON/ RAC/ VRAC(13)
COMMON/ SIZE/ NEVS
C C FOLLOWING IS RCGRAP VERSION DATE ... KEEP CURRENT!!!
64 IVER=84100
65 MCINIT=0
66 IF(LUC(-25).EG.1) MCINIT=LUC(-23)
67 CALL CALDAY(IVER,IVY,IVM,IVC,IVMC)
68 CALL ENKODE("(132X,T1,"BEGIN + VTPW + VERSION CF ",
69 12,1X,A4,I2," AT INIT ",I2")",LEUF,IVC,IVMCIY,MCINIT)
70 C C C C
71 C C LOOK FOR DIGITAL AREA NUMBER
72 NARA=IKWP(*ARA*,1,0)
73 IF(NARA.CT..MAXAPA) GC TO 900
74 C C CHECK FOR OVERRIDE CF TERMINAL NUMBER
75 IDEF=LUC(-20)
76 ITERY=IKWP(*TER*,1,IDEF)
77 C C READ *VASTEXT* OCCUPMENT RECORD & EXTRACT NEEDED INFO
78 CALL VRTIC(IRET,0,0)
79 JCAY=IDCC(1)
80 C C C C C C C C C C C C C C C C C C C C C C C C C C C C
81 JTIME=IDCC(2)
82 NENG=IDCC(38)
83 NDNP=IDCC(40)
84 NERR=IDCC(41)
85 C CHECK FOR *NC-CLPUT-MEDIUM*
86 IF(NARA.EQ.0 .AND. NERR.LT.0) GC TO 920
87 LPC=IDCC(100)
88 CGES=CKWP(*ESS*,1,0)
89 IGES=0
90 IF(CGES.EQ.CELNK) GC TO 100
91 IF(CGES.EQ.CLETG) GC TO 100
92 IF(CGES.EQ.CLETG) IGES=1
93 100 CONTINUE
94 IIFILL=0
95 IIDEF=0
96 NCSFC=IKWP(*SCF*,1,IDEF)
97 IDEF=11
98 NBXS=IKWP(*ECX*,1,IDEF)
99 IF(MOD(NBXS,2).EQ.0) IIFILL=1
100 IF(MOD(NBXS,2).EQ.0) NBXS=NBXS-1
101 NBXS=MING(NBXS,11)
102 IPGX=1
103 IF(NBXS.EQ.0) IECX=0
104 IDEF=NBXS
105 INCFL=IKWF(*SCF,1,IEF)
106 INCRE=IKWF(*SCF,2,IEF)
107 C
108 LLINE=IKWF(*END,1,IDOC(55))
109 LELEM=IKWF(*END,2,IDOC(56))
110 ILINE=IKWF(*ECC,1,IDOC(57))
111 IELEM=IKWF(*ECC,2,IDOC(58))
112 C
113 LPRUG=IKWF(*ELC,1,0)
114 LCRT=IKWF(*CRT,1,0)
115 IFLT=IKWF(*FLT,1,0)
116 KTCH=IKWF(*TCH,1,6)
117 NLPI=NL+1
118 IFRM=LUC(-1)
119 INRAS=LUC(-11)
120 INPIC=LUC(-12)
121 IF(MDRR,LT,0) GO TO 120
122 MTH=IKWF(*MTH,1,2)
123 IRET(1)=JDATE
124 IRET(2)=JTIME
125 IRET(3)=LRC
126 ICK=MPUT(*MCR,MDRR,0,IRET)
127 IF(IOK,NE,0) GO TO 880
128 120 IF(ILINE,IELEM,EG,0) GO TO 140
129 IF(LUC(16),EG,0) CC TO 180
130 CALL GETFRM(IFRM,ICIP)
131 JSAT=ICIR(1)
132 JDATE=ICIP(2)
133 CC TO 160
134 140 CALL TVSAT(IFRM,INRAS,INPIC,ILINE,IELEM,JSAT,JDATE,TIME)
135 C NEED *OLDSTYLE* JCAY FOR NAVIGATION WITHIN VASDAT
136 160 JCAY=JSAT 1CCOC+JCATE
137 180 IF((LLINE,IELEM),NE,0) GO TO 200
138 LLINE=ILINE
139 LELEM=IELEM
140 200 CONTINUE
141 VDAT(1)=-1.
142 IILINE=ILINE
143 IELEM=IELEM
144 C VASDAT CHANGES ARGUMENTS *ILINE* AND *IELEM*
145 CALL VASDAT(ILINE,IELEM,VDAT)
146 VDAT(1)=VMISE
147 JSAT=ICIR(3)
148 ISAT=ISATN(V,SAT)
149 JDAT=IDIR(4)
150 JDATE=JSAT 1CCOC+JDAT
151 JTIME=IDIR(5)
152 IRES=ICIR(12)
153 IERES=ICIR(13)
154 C SET DEFAULT TO SQUARE SAMPLES...IE EQLATE RESOLLTIONS
155 INC=IRES/IERES
156 INCIL=IRES INCRL
IF(INCIL.EQ.0) INCIL=1
INCIE=IERES INCFF=INC
IF(INCIF.EQ.0) INCIF=1
MAG2=MAG/2
MAG=MAX(0,MAG*6)
IF(KARA.EQ.0) GC TO 220

C
C SET UP AREA AND ASSOCIATED PARAMETERS
INCL=ILRES
INCEA=IERES INC
NBX=5*NX/2
ILINA=ILINE+NBX+INCLA
IELEA=IELEK+NBX+INCEA
LLINA=LLINE+NBX+INCLA
LELEA=LELEM+NBX+INCEA

C
C EMPIRICAL ACCESTMENT!
ILINA=ILINA-1
LLINA=LLINA-1

C
MLINA=(LLINA-ILINA)/INCLA+1
MELE=(LELEA-IELEA)/INCEA+1
IF(MELE.GT.660) GC TO 40
CALL ARASIZ(KARA,MLINA,MELE)
CALL ENAREA(KARA,SAT,JDATE,JTIME,ILINA,IELEA,INCLA,INCEA,IANES)
CALL OPNA(KARA)
KINES=NBX
KELEMS=NBX
KBOXES=MELE/KELEMS
IF(MCD(MELE,KELEMS).LT.0) KBOXES=KBOXES+1
IAREC=0
IF(LCRT.EQ.0) GC TO 220
CALL ENKODE("AREA ",14," OPENED WITH NL ",14,", NE ",14)
LEUF=KARA-MLINA-MELE
CALL VTG(LEUF)
220 CONTINUE

C
NDCME=0
LINET=-1
DO 820 IL=ILINE,LLINE,INCIL
LINET=LINET+1
820 IF(LCRT.EQ.0) GC TO 240
CALL ENKODE("BEGIN LINE ",14,"LEUF,IL")
CALL VTG(LEUF)
240 LASLIN=-1
KBOY=0

C
DO 740 IE=IELEM,LELEM,INCIE
KBOX=KBOX+1
LASELE=-1
IGFLG=1
ITFLG=1
DC E40 KLIN=1,KLINES
DC E20 KELE=1,KELEKS
ILFET=MAX&W
DC 260 I=4,LEAV
260 IRET(I)=VMISC
DC 280 I=1,13
VDAT(I)=VMISC
280 MISP(I)=0
ILINES=IL
IELES=IE
CALL VASD(T(ILINES,IELES,VDAT))
IFAIL=1
IF(ABS(VLATS)GE.60.) GO TO 600
IF(ABS(VZEN)GT.60.) GO TO 600
IF(LELEV*59.9999) GO TO 600
IF(VDAT(I).LT.0.) GO TO 600
IMF=1
300 CONTINUE
MSAM=0
DO 320 I=1,12
IF(VDAT(I).LT.180.)GO TO 320
IF(VDAT(I)EG.VMISC) GO TO 320
IF(VDAT(I).GT.330.) GO TO 320
IF(VDAT(I).LT.330.) GO TO 320
320 CONTINUE
320 CONTINUE
IFAIL=2
IF(MSAM.EG.0.) GO TO 600
IF(IGFLG.0.) GO TO 480
CALL GESPRC(IGES,NCSCF,MDNG)
IFAIL=3
IF(TGES(1).LE.0.) GO TO 600
IGFLG=0
PSTA=IPSTA
TSTA=.01 FLCAT(ITSF)
TDSTA=0.01 FLCAT(ICSF)
E=SATVAP(TTESTA)
WSTA=(E22.1E)/(PSTA-E)
LS=NL
DG 360 J=1,20
I=NLP1-1
DF=F(I)-PSTA
IF(DP.GT.0.) GO TO 360
IS=I+1
IS=MINS(IS,NL)
GC TO 390
CONTINUE
390 NLS=MINC(IS, LS)
DTS=TSTA-T(NLS)
ILC=37
IF(IGES.EQ.1) ILC=26
IF(ILC.GE.NLS) ILC=ILC-3
DTCF=DTS/(F(NLS)-F(ILC))
CC 440 I=1, 40
IF(I.GE.20) GC TO 400
W(I)=.02
GC TO 440
CONTINUE
400 GC TO 420
E=SATVAP(T(I))
W(I)=.2*(622 E/(P(I)-E))
GC TO 440
420 W(I)=W(I)*WSTAB/W(NLS)
T(I)=T(I)+CTCF*(P(I)-P(ILC))
IF(I.LE.NLS) GC TO 440
T(I)=TSTA
W(I)=WSTAB
CONTINUE
440 CONTINUE
460 CONTINUE
IF(LBUG.EQ.0) GC TO 480
CALL ENKCCF(*12X,T1,5X,"PRESSURE ", 15F7.1/), LBUF,F(26)
CALL ENKCCF(*5X,"GUESS TEMP", 15F7.1/), LBUF,T(26)
CALL ENKCCF(*5X,"GUESS WMR", 15F7.3/), LBUF,W(26)
CALL ENKCCF(*1X,"Z1000 =","I8,5X,"TSFC =","I8,5X,"TDSFC =","I8")
+ LELF,IZ16,ITSFC,ICSFC)
480 CONTINUE
LAT=IRCOUNT(VLAT, 10C.)
LCM=IRCOUNT(VLCNS 10C.)
IVZ=IRCOUNT(VZEN 10C.)
ISZ=IRCOUNT(SZEN, 10C.)
IF(LBUG.EQ.0) GC TO 500
LBUF)
CALL LTG(LBUF)
CALL ENKCCF(*12X,T1,6I8)*LEUF,IL,IE,LAT,LCM,IVZ,ISZ)
CALL LTG(LBUF)
CALL VTG(LBUF)
CALL SDEST(* SAMPLE = *,MSAM)
500 CONTINUE
CALL ACQUISITION COMPLETE
CALL VASTFLK(VCAT,TCCT,TSFS,LRST,ITFLG,ISAT,KTCH,NLS,STAB)
IFAIL=7
IF(UPET.EQ.WISE) GC TO 600
IFAIL=0
KFLAG=2

483
C   OUTPUT ONLY EVERY *NTH* RETRIEVAL TO MD-FILE
MDCN=MDCK+1
MCDLT=C
IF(MDCN(.NE.NTH).AND.MEQ.1) MDCLT=1
IF(NTH.EQ.1) MDCLT=1
C   MAXIMUM PERMISSIBLE VALUE OF *URET* IS 12 CM
IURET=IRCLAC(URET=2G.)
C
IF(MDRR.LT.C) GC TO 540
IF(MDCLT.EQ.0) GC TO 540
K=10
DC 520 I=KTCP+P
IF(KTCP.EQ.0.AND.I.EQ.0) GC TO 520
K=K+1
IF(VDAT(I).EQ.VMIST) GO TO 520
IRET(K)=IRCLAC(VCAT(I)=100.)
520 CONTINUE
IRET(14)=IRCLAC(TSFS=100.)
540 IF(LBUC.EQ.0) GO TO 560
CALL ENKTS*((132*11,1,4)**+T-P-W = "*FE.2",LEUF,URET)
IF(LCRT.LEC) CALL VTMG(LEUF)
IF(LBUC.LEC) CALL LTG(LEUF)
560 CONTINUE
IF(IPLT.LEC) GC TO 580
CALL SATTV(IFRM,ILINES,IELES,IRAS,IFIC,JSAT,JDAY,JTIME)
JDAY=JSAT 100000+JDAY
IUR=IRCLAC(URET=10.)
KCOLOR=KFLAG
CALL VASDGC(IRAS+MAG2,IFIC+MAG2,IUR,MAG,1,KOLOR)
580 CONTINUE
IF("DRR.LT.0) GC TO 600
IF(MDCLT.LEC) GO TO 600
IPET(4)=2-KFLAG
IPET(5)=IRCLAC(VLATS=10000.)
IPET(6)=IRCLAC(VLONS=10000.)
IPET(7)=IRCLAC(VZEM=10000.)
IPET(9)=ILINES
IPET(5)=IELES
IPET(10)=KFALL
IPET(15)=IRCLAC(URET=1000.)
IPET(16)=IRCLAC(STAE=1000.)
LRO=LRC+1
ICK=MDCPUT(MCNR,MDRR,LRC,IPET)
IF(IOK.NE.0) GO TO 860
IOP=127
IOP=MDCPUT(MCNR,MDRR,LRC,IPET)
600 CONTINUE
IF(IFAIL.NE.0) NFAIL(IFAIL)=NFAIL(IFAIL)+1
IF(NARA.EQ.0) GC TO 620
VAS8(KELE,KLIN)=VDAT(8)
FARRAY(KELE,KLIN)=URET
484
C
LFLAG(KELE,KLIN)= IFAIL
IARRAY(KELE,KLIN,KECX)= IURET
C
620 CONTINUE
640 CONTINUE
371 IF(IFILL .NE. 0) GO TO 720
372 SRATIC=0.
373 XMIN=VMISG
374 DC 700 IIE=1,KELEMS
375 RATIC(IIE)=0.
376 S=0.
377 DC 680 IIL=1,KLINES
378 IF(LFLAG(IIE,IIL) .LE. 7) GO TO 680
379 X=VAS8(IIE,IIL)
380 Y=FARRAY(IIE,IIL)
381 IF(X .LE. YMISG .OR. Y .LE. VMISG) GO TO 660
382 X=X-TSTA+3.
383 IF(X .LE. 0.) GO TO 660
384 XMIN=AMIN1(XMIN,X)
385 RATIC(IIE)=Y*X/RATIC(IIE)
386 S=S+1.
387 SRATIO=RATIC(IIE)/S
388 660 CONTINUE
389 IF(SRATIO .GE. .0.) GO TO 680
390 IF(LFLAG(IIE,IIL) .LE. 7) GO TO 680
391 Y=FARRAY(IIE,IIL)
392 IF(Y .LE. VMISG) GO TO 690
393 X=VAS8(IIE,IIL)
394 IF(X .LE. VMISG) GO TO 690
395 X=X-TSTA+3.
396 IF(X .LE. 0.) GO TO 680
397 IF(Y .LE. XMIN) GO TO 680
398 Y=SRATIO*X
399 IARRAY(IIE,IIL,KECX)=IRCUNE(Y+20.)
400 680 CONTINUE
401 700 CONTINUE
402 720 CONTINUE
403 740 CONTINUE
404 IF(NARA .EQ. 0) GO TO 800
405 C
406 DC 780 J=1,KLINES
407 IF=0
408 DO 760 K=1,KEYES
409 DC 760 I=1,KELEMS
410 IA=IA+1
411 IARET(IA)=IARRAY(I,J,K)
412 760 CONTINUE
413 CALL PACM(KELE,IARET,IAELF)
414 CALL WRITA(NAPA,IAREC,IABUF)
415 IAREC=IAREC+1
416 780 CONTINUE
800 CALL EDEST(*FINISHED LINE*,IL)
820 CONTINUE
420 IF(NDRR.LT.0) GO TO 840
420 ICCC(100) = LRC
422 CALL VRTIC(IRET*0+1)
423 840 IF(NARA.NE.0) CALL CLOSAC(NARA)
424 CALL EDEST(*: ALL DONE :..:0)
425 CALL ENKCCF(*("FAILRE SUMMARY ",715),LEUF,FAIL)
426 CALL VTG(LELF)
427 RETURN
428 860 CALL EDEST(*TACLELE WRITING COLUM\(RECORD) ,LRC)
429 GO TO 960
430 880 CALL EDEST(*TACLELE WRITING HEADER FOR ROW ,MCRR)
431 GO TO 960
432 900 CALL EDEST(*INVALID AREA NUMBER ,NARA)
433 GO TO 960
434 920 CALL EDEST(*NO OUTPUT MEDIUM (AREA OR MC) SPECIFIED :..:0)
435 GO TO 960
436 940 CALL EDEST(*IMAGE TCC LIKE - MAXLE IS 660*,E512 = ,MELE)
437 950 CALL ABORT(C)
438 RETURN
439 END
References

