DTIC FILE COPY

(2)

MEMORANDUM REPORT BRL-MR-3896

BRL

AD-A232 562

PROGRAMMER/ANALYST GUIDE FOR THE ARMY UNIT RESILIENCY ANALYSIS (AURA) AUXILIARY CODE: AURATEK

PATRICK M. SWOBODA

DTIC
ELECTE
MAR 12 1991
S B D

FEBRUARY 1991

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

U.S. ARMY LABORATORY COMMAND

BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

91 3 06 002

NOTICES

Destroy this report when it is no longer needed. DO NOT return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1991	Final, 2 Jul 90 - 2 Jan 91

4. TITLE AND SUBTITLE

Programmer/Analyst Guide for the Army Unit Resiliency Analysis (AURA) Auxiliary Code: AURATEK

5. FUNDING NUMBERS

DA31 6146
1L162618AH80

6. AUTHOR(S)

Patrick M. Swoboda

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-5066

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

BRL-MR-3896

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

AURATEK is an auxiliary computer code used to assist with Army Unit Resiliency Analysis (AURA) methodology analyses. This program provides a number of services to the user. First, the program will generate a new unit deployment or modify an existing one. In addition, the program provides a graphical representation of the unit deployment and threat, along with the option to manipulate the deployment using the Precision Visuals Inc. DI-3000 software package. Finally, the program can provide a graphical output of a weapon laydown used in an AURA simulation.

14. SUBJECT TERMS

AURA
AURATEK
Auxiliary
Deployment
Graphics
Output

15. NUMBER OF PAGES

129

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

INTENTIONALLY LEFT BLANK.

CONTENTS

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

INTENTIONALLY LEFT BLANK.

I. INTRODUCTION

AURATEK is an auxiliary computer code used to assist with Army Unit Resiliency Analysis (AURA) methodology analyses. AURATEK was designed to be used as an interactive tool, which graphically represents the deployment of a unit in order to provide the AURA analyst with the capability to verify and/or modify the deployment section of the AURA input file.[1] A detailed description of the AURA model is contained in a two volume BRL report.[2,3] Consequently, a detailed discussion of AURA inputs will not be presented in this report.

The AURATEK program provides a number of services to the user. First, the program will generate a new AURA deployment or modify an existing deployment. In addition, the program provides a graphical representation of the unit deployment and threat, along with the option to manipulate the deployment using the Precision Visuals Inc. DI-3000 software package. Finally, the program can provide a graphical output of a weapon laydown used in an AURA simulation. The program has a main menu of options and a weapon mode with options corresponding to the weapon deployment.

II. EXECUTION OF AURATEK

1. Operation of AURATEK

AURATEK is a Fortran code currently configured to be used with the Precision Visuals Inc. DI-3000 commands. To execute AURATEK the user types "tek". When first entering the program, the user has the option to input an existing deployment by responding with the name of the input file or create a new deployment by responding with a "0". This input file is the same format as the AURA deployment file. Next the user is prompted if there is symbol/color data to be read from that input file. This data would have been appended to the end of the deployment used in AURATEK. This allows the user to represent a certain

---

1. Klopcic, T.J., "Input Manual for the Army Resiliency Analysis (AURA) Methodology: 1988 Update", U.S. Army Ballistic Research Laboratory, BRL-TR-2914, May 1988, (UNCLASSIFIED)

2. Sheroke, Robert M. et. al., "Programmer/Analyst Guide for the Army Unit Resiliency Analysis (AURA) Computer Simulation Model, Volume 1: AURA Methodology", U.S. Army Ballistic Research Laboratory, BRL-TR-3156, August 1990, (UNCLASSIFIED)

3. Sheroke, Robert M. et. al., "Programmer/Analyst Guide for the Army Unit Resiliency Analysis (AURA) Computer Simulation Model, Volume 2: AURA Source Code", U.S. Army Ballistic Research Laboratory, BRL-TR-3103, July 1990, (UNCLASSIFIED)

asset within a deployment with a specific symbol or color. The next question asks if the user wants hints on using AURATEK; this "Helpful hints" section gives some hints on command input and cursor control in AURATEK. The program then reads in the deployment and plots it on the screen with an option menu. If symbol/color data were used the corresponding assets would be plotted with the associated symbol/color. The user proceeds with the AURATEK program by selecting an option from the menu with which to modify the existing plot.

2. AURATEK Inputs/Outputs

AURATEK requires one input file, specifically, an AURA input file or a zero for a new file. A detailed discussion of an AURA deployment can be found in references 1 and 2. However, an example of a deployment input follows:

```
DEPLOYMENT
AIRCRAFT,    -280.,1520.,      1.,1,1,1,1,1,5
AIRCRAFT,    -320.,1570.,      1.,1,1,1,1,1,5
END
```

The first input is the asset name at that target point. The next two inputs are x and y coordinates, respectively. The fourth input is the number of assets at that point. The fifth, sixth and seventh inputs are the conventional, nuclear, and toxic kill criteria codes, respectively. The final three inputs are conventional, nuclear, and toxic posture codes.

If the user wishes to view an AURA threat weapon laydown for a given deployment, a second input file is required. This file is created by using the DUMP9 option in AURA; DUMP9 will create the weapon file for a conventional, chemical or nuclear laydown. Note, this file is only needed if the weapon option in AURATEK is used; AURATEK does not require it. Figure 1 illustrates the symbols used to represent the three weapon types available: chemical, nuclear and conventional. A chemical munition is represented by an asterisk which denotes its function point and further characterized an ellipse which designates the orientation of the chemical dissemination pattern on the ground. A nuclear weapon function point is simply characterized by an asterisk. A conventional weapon function point is characterized by an asterisk with a circle around it.

AURATEK produces two output files. The first is the deployment file which the user has the option of saving if a deployment was modified or created. AURATEK prompts the user for a number to name the file. It is then named fort.xx, where xx is the prompted number. Also the user has the option to append to the

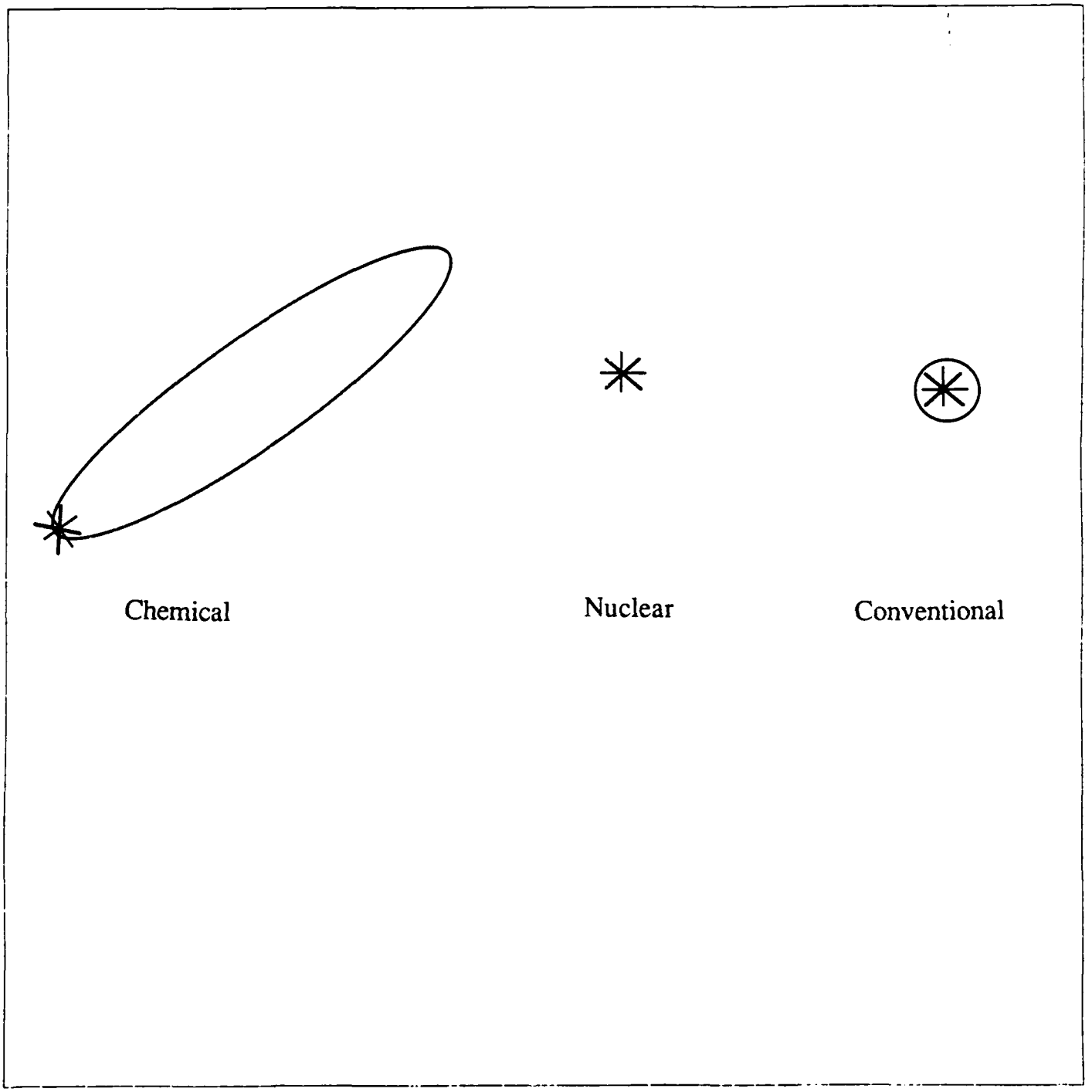

Figure 1. Example of chemical, nuclear, and conventional weapon laydown.

end of the deployment file, codes corresponding to symbols and colors assigned to different assets in the deployment. Below is an example of a set of values appended to a deployment file:

AIRCRAFT	2	1
POL TRUCK	1	2
PILOT	4	3

The first column is the asset name, the second column corresponds to the symbol, and the third column corresponds to the color. This appended section can only be used within AURATEK and must be deleted when used as a deployment in AURA.

The second device variable in the user environment sets the hardcopy file that contains all the plots produced by AURATEK for the given session. A new plot is created with each option used in AURATEK. Figure 4 is an example of a hardcopy plot produced by AURATEK. This example uses the appended symbol information to denote different types of assets on the plot. PVI automatically writes over the file created in the previous PVI session.

3. AURATEK User Options

Options are provided through the course of program execution in the form of a menu listing. The following is the current menu of options and descriptions available in the AURATEK program:

COORDS	Enables user to determine the X, Y coordinates at cursor location.
DEPLOY	Enables the user to indicate a location on the screen and deploy an asset at that point. Default values can be used for kill criteria and postures, if desired.
DISTANCE	Enables the user to determine the distance between any two points on the screen.
END	Causes the program to terminate. User may designate what file is to be used for the output or enter "0" to indicate no output file is desired.
FULL	Restores the screen to its initial dimensions.
GRID	Draws axes on the bottom and left side of the screen and prints out the tic interval.

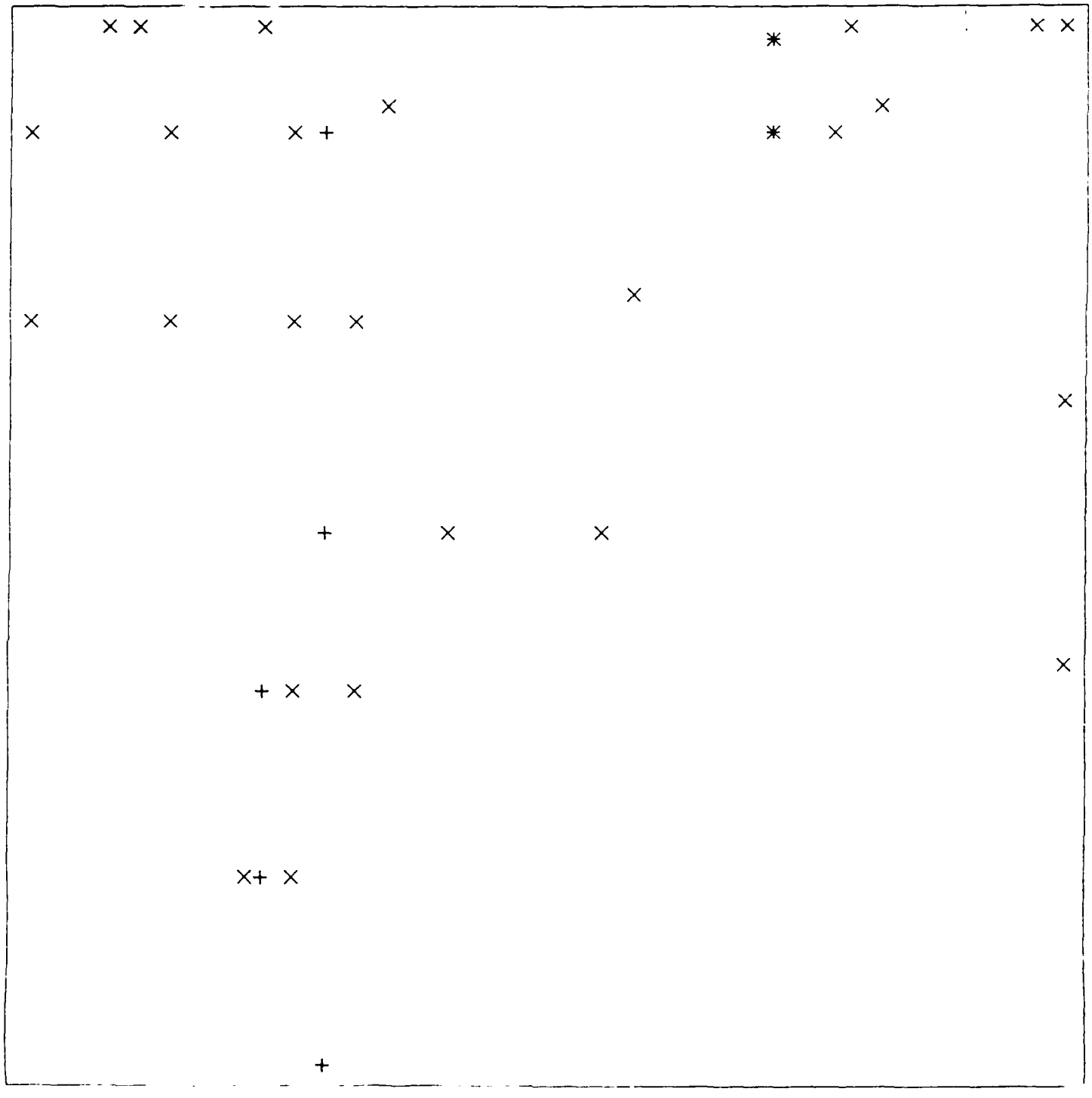

Figure 2. AURATEK deployment plot.

INPUT	Used in weapon mode to read in the threat data (DUMP9 file) from the tape specified by the user. Prompts the user for the data file to display. The INPUT command also rescales screen ensuring that none of the weapon patterns overlap the screen.
BACK	Returns the user to the normal mode. Used in weapon mode.
EMPLOY	Used within weapon mode to employ weapon patterns on the screen.
HELP	Provides a brief description of each command. Format for help is: Help "Command name".
IDENTIFY	Identifies the asset at the location specified by the cursors. If in weapon mode, identifies weapon at the cursor.
LIST	Lists all weapon numbers and names.
LOCATE	Locates a given X, Y coordinate on the screen and places a "+" there.
MENU	Prints the list of available commands.
MOVE	Moves or removes the indicated asset.
REFRESH	Redraws current screen.
SCREEN	Displays the current screen dimension.
SET	Enables user to set the minimum and maximum values for screen dimensions.
SYMBOL	Enables user to designate different symbols for asset types.
WEAPON	Enables user to enter weapon mode.
ZOOM	Expands a part of the current screen.

III. COMPUTER REQUIREMENTS

Program AURATEK is written in FORTRAN 77 and conforms to the conventions of the ANSI FORTRAN 77 complier. The Ballistic Research Laboratory maintains and executes AURATEK on a Gould-UTX/32 running the UNIX 5.0 operating system with Precision Visuals Inc. graphics capibilities.

IV. AURATEK Subroutines

Appendix A contains an alphabetized collection of the subroutines which comprise the AURATEK source code. Included for each subroutine are both a subroutine description report and the FORTRAN source code. The description report contains the following information, in detail:

1) Function of subroutine;
2) Required subroutine parameters;
3) Discussion of subroutine methodology and processing sequence;
4) Subroutine program variable glossary including:

- Variable name,
- Variable definition,
- Type (Real, Integer, etc),
- Units (Minutes, Meters),

5) Subroutine cross referencing data including:

- Program subroutines called by subroutine;
- Standard FORTRAN 77 Library functions used;
- Parent subroutine(s) (i.e. subroutine(s) from which it is called.)

Appendix B contains an alphabetized glossary of AURATEK program variable definitions. The AURATEK program variable glossary includes all global program variables used within the AURATEK source code.

V. SUMMARY

AURATEK is an auxiliary Fortran computer code used to graphically assist with the deployment of targets for the Army Unit Resiliency Analysis. AURATEK also provides the analyst with a graphical display of the weapon laydown. The purpose of this report is to assist in the operation of AURATEK. AURATEK allows the user to graphically display and create new deployments.

INTENTIONALLY LEFT BLANK.

APPENDIX A

AURATEK Subroutines

INTENTIONALLY LEFT BLANK.

SUBROUTINE AURATEK

FUNCTION: Subroutine AURATEK initializes variables and reads in an AURA deployment file.

PARAMETERS: NONE

DISCUSSION: Subroutine AURATEK is called by MAIN to initialize variables and read in an AURA deployment file. This section reads the deployment in as a string then divides the string into proper postures for that asset. Also contained in this subroutine is the "Helpful Hints" section which gives the first time user information about the menu input and cursor control.

AURATEK VARIABLE GLOSSARY:

Variable	Definition	Type	Units
ALTPOS	Holds alternate posture values.	Char.	N/A
BORDER	Turns border on or off around the work area.	Logical	N/A
COMM	Holds commented deployment line.	Char.	N/A
ICOLOR	Sets color for symbol read from the input file.	Char.	N/A
ILINE	Reads in each line of deployment.	Char.	N/A
IPAR	Array of kill criteria & initial postures.	Integer	N/A
ISYM	Sets PVI symbol value written to the output file.	Integer	N/A
IWORD	Holds first word of deployment line.	Char.	N/A
L7	Used to flag end of data in ILINE.	Logical	N/A
MOPP	Used to flag "MOPP ALL" line in the input file.	Logical	N/A
MOPPALL	Equals ILINE if 'MOPP ALL' is in deployment.	Char.	N/A
NAME	Asset name.	Char.	N/A
NDEPL	Number of deployment points.	Integer	N/A
N1	Keeps count of commas separating target description.	Integer	N/A
OFFSET	Holds offset value of deployment. Used if assets are duplicated within	Real	N/A

	the deployment file.		
PAR	Array holding number of assets with the same name at a given target point.	Real	N/A
XTAR	Array of x and y coordinates of asset deployment.	Real	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: EXIT

System Library functions: NONE

Subroutines calling AURATEK: MAIN

```
C*****************************AURATEK*************************************
      SUBROUTINE AURATEK
C
C  THIS PROGRAM IS A GRAPHIC SUPPORT PACKAGE FOR AURA.  IT READS
C  IN AN AURA DEPLOYMENT FILE (IF AVAILABLE), DISPLAYS THE X,Y
C  COORDINATE GRID AND ALLOWS THE USER TO MANIPULATE THE TARGETS
C  ON THE SCREEN TO ADJUST OR CREATE A DEPLOYMENT FILE.
C  ADDITIONAL CAPABILITIES ALLOW THE USER TO READ IN WEAPON
C  DATA FROM AN AURA RUN AND PLOT THE WEAPON EMPLOYMENT PATTERNS.
C
C
C  INITIALIZE NDEPMX TO MAX. NUMBER OF DEPLOYMENT POINTS.
C
      PARAMETER ( NDEPMX = 702 )
      COMMON/BRDR/ BORDER
      COMMON/NAMES/NAME(NDEPMX),ISYM(NDEPMX),ICOLOR(NDEPMX),
     + ALTPOS(NDEPMX), COMM(NDEPMX)
      COMMON/DEPLOY/ NDEPL, XTAR(NDEPMX,2), IPAR(NDEPMX,7)
      COMMON/NERD/ MOPP,OFFSET(2)
      COMMON/HNERD/ MOPPALL
      DIMENSION PAR(NDEPMX,7)
      EQUIVALENCE ( PAR,IPAR )
      CHARACTER*18 IWORD,ILINE*80,NAME,YN*1,ISYM*1,ICH*1,ICOLOR*1,ICOL*1
     +, FNAME*8, ALTPOS*40, COMM*40, MOPPALL*80, ANS*1
      LOGICAL L7, MOPP, BORDER
C
C  MOPP IS A LOGICAL VARIABLE WHICH FLAGS THE "MOPP ALL" LINE
C  IN THE INPUT FILE.
C
      MOPP = .FALSE.
C
C  BORDER IS A LOGICAL VARIABLE WHICH FLAGS WHETHER OR NOT TO
C  INCLUDE A BORDER AROUND THE WORK AREA.
C
      BORDER = .FALSE.
C
C  INITIALIZE ARRAYS
C
      DO 105  J = 1, NDEPMX
        ALTPOS(J) = ' '
        COMM(J) = ' '
        ICOLOR(J) = ' '
        ISYM(J) = 'X'
  105 CONTINUE
      OFFSET(1) = 0.0
      OFFSET(2) = 0.0
C
C  WELCOME USER TO AURATEK.  SHORT INTRODUCTION.
C
      WRITE(6,106)
  106 FORMAT(70('*'),/,'*',68X,'*',/,'*',3X,'WELCOME TO AURATEK!  AURA',
     + 'TEK IS AN ',
     + 'INTERACTIVE GRAPHICS ',9X,'*',/,'*',3X,'PROGRAM WHICH WILL ',
```

```
     + 'ASSIST YOU IN DEVELOPING AND CORRECTING AURA',2X,
     + '*',/,'*',3X,'DEPLOYMENTS.  AURATEK ALSO ASSISTS IN CHECKING ',
     + 'THE RESULTS OF ',3X,'*',/,'*',3X,
     + 'WEAPONS EMPLOYMENT.',46X,'*',/,'*',68X,'*',/,70('*'),/)
C
C  INTERACTIVELY ASK USER FOR THE NAME OF THE DEPLOYMENT
C  FILE AND IF SYMBOLS ARE INCLUDED WITH THE DATA
C
      WRITE(6,109)
  109 FORMAT(' ENTER NAME OF LOCAL INPUT FILE (IF NONE, ENTER 0) >  :')
      READ(5,160) FNAME
  160 FORMAT(A8)
      IF( FNAME .EQ. '0') GO TO 200
      OPEN (10,FILE=FNAME)
      REWIND 10
      DO 110 I = 1, 3000
        READ(10,119,END=115) IWORD
  119 FORMAT( A12 )
        IF( IWORD(1:6) .EQ. 'DEPLOY' )  GO TO 120
  110 CONTINUE
  115 WRITE(6,1119)
 1119 FORMAT(' COULD NOT FIND DEPLOYMENT ON INPUT FILE' )
      STOP1
  120 NDEPL = 0
C
C  STARTS READING IN DEPLOYMENT INFORMATION.
C
  125 READ(10,129,END=200) ILINE
  129 FORMAT( A80 )
C
C  CHECKS TO SEE WHAT TYPE OF INFORMATION ILINE HAS IN IT.
C
      IF( ILINE .EQ. 'END' )  GO TO 190
      IF (ILINE(1:1) .EQ. '#') THEN
        GO TO 125
      ELSE IF (ILINE(1:1) .EQ. '$') THEN
        IF (NDEPL .NE. 0) THEN
          ALTPOS(NDEPL) = ILINE(:40)
        ELSE
          GO TO 125
        ENDIF
      ELSE IF (ILINE(:8) .EQ. 'MOPP ALL') THEN
        MOPP = .TRUE.
        MOPPALL = ILINE
      ELSE IF (ILINE(:7) .EQ. '&OFFSET') THEN
        N1 = INDEX(ILINE,',')
        IF (N1 .GT. 0) THEN
          ILINE = ILINE(N1+1:)
          N1 = INDEX(ILINE,',')
          IF (N1 .GT. 0) THEN
            IWORD = ILINE(:N1-1)
            READ(IWORD,139) OFFSET(1)
  139 FORMAT( F18.3 )
```

```
          IWORD = ILINE(N1+1:N1+19)
          READ(IWORD,139) OFFSET(2)
        ELSE
          WRITE(6,141)
 141 FORMAT('** ERROR ON "OFFSET" LINE!!!  CONTINUE? > :')
          READ(5,2059) ANS
2059 FORMAT( A1)
          IF (ANS .NE. 'Y') STOP2
          PRINT*,'OFFSET LINE IGNORED.  PROGRAM CONTINUING ... '
        ENDIF
      ELSE
        WRITE(6,141)
        READ(5,2059) ANS
        IF (ANS .NE. 'Y') STOP2
        PRINT*,'OFFSET LINE IGNORED.  PROGRAM CONTINUING ... '
      ENDIF
    ELSE
      N1 = INDEX(ILINE,'#')
      IF (N1 .GT. 0) THEN
        COMM(NDEPL) = ILINE(N1:N1+40-1)
        ILINE = ILINE(:N1-1)
      ENDIF
      IF( ILINE .EQ. 'END' )  GO TO 190
      N1 = INDEX(ILINE,',')
      IF( N1 .EQ. 0 )  THEN
        WRITE(6,7739) NDEPL
7739 FORMAT(' ERROR IN INPUT FILE ON ITEM NO. ', I6, /
   + ' NOTE: FOR AURATEK, DEPLOYMENT MUST END WITH "END" CARD')
        CALL EXIT
      ENDIF
C
C  TAKES DEPLOYMENT LINE AND SEPARATES INTO POSTURES.
C  DEPLOYMENT IS INITIALLY READ IN AS A CHARACTER. THIS SECTION
C  SEPARATES THE CHARACTERS INTO THE APPROPRIATE NUMBERED
C  POSTURES.
C
      NDEPL = NDEPL+1
      IF( NDEPL .LE. NDEPMX )  GO TO 127
      WRITE(6,1229) NDEPMX
1229 FORMAT(' DEPLOYMENT FILE CONTAINS OVER ',I4,
   +' EXCEEDS STORAGE' )
      STOP1
 127  NAME(NDEPL) = ILINE(:N1-1)
      ILINE = ILINE(N1+1:)
      DO 140  J = 1, 2
        N1 = INDEX(ILINE,',')
        IF( N1 .LE. 0 )  THEN
          WRITE(6,7739) NDEPL
          CALL EXIT
        ENDIF
        IWORD = ILINE(:N1-1)
        READ(IWORD,139) XTAR(NDEPL,J)
        XTAR(NDEPL,J) = XTAR(NDEPL,J) + OFFSET(J)
```

```
        ILINE = ILINE(N1+1:)
  140   CONTINUE
        L7 = .TRUE.
        DO 150  J = 1, 6
          N1 = INDEX(ILINE,',')
          IF( N1 .LE. 0 )  THEN
            IF( J .LT. 5 )  THEN
              WRITE(6,7739) NDEPL
              CALL EXIT
            ENDIF
            IPAR(NDEPL,7) = 0
            IPAR(NDEPL,5) = IPAR(NDEPL,4)
            IPAR(NDEPL,4) = 1
            L7 = .FALSE.
            GO TO 152
          ENDIF
          IWORD = ILINE(:N1-1)
          N0 = INDEX(IWORD,'.')
          IF( N0 .LE. 0 )  THEN
            READ(IWORD,159) IPAR(NDEPL,J)
  159 FORMAT( BN,I18 )
            IF( J .EQ. 1 )  PAR(NDEPL,1) = FLOAT(IPAR(NDEPL,1))
          ELSE
            READ(IWORD,139) PAR(NDEPL,J)
          ENDIF
          ILINE = ILINE(N1+1:)
  150   CONTINUE
  152   CONTINUE
        READ(ILINE,159) IPP
        IF( L7 )  IPAR(NDEPL,7) = IPP
        IF( .NOT. L7 )  IPAR(NDEPL,6) = IPP
      ENDIF
      GO TO 125
  190 CONTINUE
C
C  IS THERE SYMBOL DATA STORED ON THE FILE?
C  THE SYMBOLS ARE STORED BY ASSET NAME ( SEE 250 LOOP BELOW ).
C
      WRITE(6,199)
  199 FORMAT(' IS THERE SYMBOL/COLOR DATA TO BE READ FROM FILE? ',
     +'Y OR N? > :' )
      READ(5,2059) YN
      IF( YN .NE. 'Y' )  GO TO 200
      DO 195  K = 1, 1000
        READ(10,1199,END=200) IWORD,ICH,ICOL
 1199 FORMAT( A18, 3X, A1, 3X, A1 )
        DO 198  I = 1, NDEPL
          IF( NAME(I) .EQ. IWORD )  ISYM(I) = ICH
          IF(NAME(I) .EQ. IWORD)  ICOLOR(I) = ICOL
  198   CONTINUE
  195 CONTINUE
  200 CONTINUE
C
```

```
C  CHECKS TO SEE IF USER WOULD LIKE BEGINNER HINTS.
C
     WRITE(6,108)
  108 FORMAT(" WOULD YOU LIKE BEGINNER'S HINTS BEFORE YOU START? >:")
     READ(5,2059) ANS
     IF (ANS .EQ. 'Y') THEN
       WRITE (6,111)
       READ(5,2059,END=112) ANS
       IF (ANS .EQ. 'N') STOP
     ENDIF
  111   FORMAT(/,'THIS PROGRAM IS BASED ON A MENU-TYPE SYSTEM.  THUS,',
    + /,'YOU WILL BE GIVEN A CHOICE OF COMMANDS TO SELECT FROM.  A',
    + /,'MINIMUM OF THREE CHARACTERS MUST BE ENTERED FOR EACH COMMAND.'
    + ,/,'WHEN ANSWERING A QUESTION, ONLY ONE CHARACTER RESPONSES ARE',
    + /,'REQUIRED.',/,
    + /,'"CURSORS" ARE USED BY MANY OF THE COMMANDS.  YOU CAN CONTROL',
    + /,'THESE CURSORS WITH THE JOY DISK LOCATED ON THE UPPER LEFT ',
    + /,'PART OF YOUR KEYBOARD.  WHEN YOU HAVE MOVED THE CURSORS SO ',
    + /,'THAT THEY INTERSECT OVER THE DESIRED POINT ON THE SCREEN ',
    + /,'PRESS ANY ALPHA NUMERIC KEYBOARD CHARACTER.  DO NOT PRESS'
    + ,' RETURN !!!!',//,
    + 'YOU ARE NOW READY TO START AURATEK!  CONTINUE? >  :')
C
C    INPUT FILE, IF ANY, IS READ IN.
C    READY TO CALL GRAPHICS PORTION OF PROGRAM.
C
  112  RETURN
     END
```

SUBROUTINE ATEKOUT

FUNCTION: Subroutine ATEKOUT is used to write a new deployment file, including assets added or removed and append the symbols and colors that AURATEK uses to the end of the file.

PARAMETERS: NONE

DISCUSSION: Subroutine ATEKOUT is used to write a deployment file for assets added or removed and append the symbols and colors that AURATEK uses to the end of the file. This subroutine rewrites the deployment with any changes the user has made to the original deployment.

ATEKOUT VARIABLE GLOSSARY:

Variable	Definition	Type	Units
ALTPOS	Holds alternate posture values.	Char.	N/A
BORDER	Turns border on or off around the work area.	Logical	N/A
COMM	Holds commented deployment line.	Char.	N/A
ICOLOR	Sets PVI color value for symbol written to the output file.	Integer	N/A
IPAR	Array of kill criteria & initial postures.	Integer	N/A
ISYM	Sets PVI symbol value written to to the output file.	Integer	N/A
MOPP	Used to flag "MOPP ALL" line in the input file.	Logical	N/A
MOPPALL	Equals ILINE if 'MOPP ALL' is in deployment.	Char.	N/A
NAME	Asset name.	Char.	N/A
NDEPL	Number of deployment points.	Integer	N/A
OFFSET	Holds offset value of deployment. Used if certain groups of assets are duplicated within the deployment file.	Real	N/A
PAR	Array holding number of assets with the same name at a target point.	Real	N/A
XTAR	Array of x and y coordinates of asset deployment.	Real	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: NONE

Subroutines calling ATEKOUT: MAIN

```
C************************** ATEKOUT ************************************

      SUBROUTINE ATEKOUT

C
C  SUBROUTINE ATEKOUT IS USED TO WRITE A NEW DEPLOYMENT FILE
C  INCLUDING ASSETS ADDED OR REMOVED AND APPEND THE SYMBOLS AND
C  COLORS THAT AURATEK USES TO THE END OF THE FILE.
C
      COMMON/BRDR/ BORDER
      COMMON/NERD/ MOPP,OFFSET(2)
      COMMON/HNERD/ MOPPALL
      COMMON/NAMES/NAME(NDEPMX),ISYM(NDEPMX),ICOLOR(NDEPMX),
     + ALTPOS(NDEPMX), COMM(NDEPMX)
      COMMON/DEPLOY/ NDEPL, XTAR(NDEPMX,2), IPAR(NDEPMX,7)
      CHARACTER*18 NAME, YN*1, ISYM*1, ICOLOR*1, ALTPOS*40, COMM*40,
     + MOPPALL*80
      PARAMETER ( NDEPMX = 702 )
      LOGICAL MOPP, BORDER
      DIMENSION PAR(NDEPMX,7), OFFSET(2)
      EQUIVALENCE ( PAR,IPAR )

C
C  THIS PART WRITES A NEW DEPLOYMENT TO A FILE.
C
      WRITE(6,209)
      READ(5,*) IO
      IF( IO .LE. 0 )  GO TO 5000
      REWIND IO
      WRITE(IO,29)
      IF (MOPP) WRITE(IO,9) MOPPALL
    9 FORMAT( A80 )
      DO 10  N = 1, NDEPL
        WRITE(IO,239) NAME(N),(XTAR(N,K),K=1,2),PAR(N,1),
     +   (IPAR(N,K),K=2,7), COMM(N-1)(:16)
        IF (ALTPOS(N-1) .NE. ' ')  WRITE(IO,279) ALTPOS(N-1)
   10 CONTINUE
      WRITE(IO,49)
C
C  APPENDS COLORS AND SYMBOLS ASSOCIATED WITH SPECIFIC ASSETS
C  TO END OF FILE FOR USE IN AURATEK.
C
      WRITE(6,59)
      READ(5,89) YN
      IF( YN .EQ. 'N' )  GO TO 5000
C
C  RUNS THROUGH EVERY ASSET NAME TO SEE WHICH ONES TO PUT IN
C  SYMBOL & COLOR FILE.
C
      DO 20  N = 1, NDEPL
        DO 30  K = 1, N-1
          IF( NAME(K) .EQ. NAME(N) )  GO TO 250
   30   CONTINUE
```

```
C
C  FORMAT FOR AURATEK INFORMATION AT THE END OF THE DEPLOYMENT
C  FILE.
C
      WRITE(IO,69) NAME(N),ISYM(N),ICOLOR(N)
  20 CONTINUE
 5000 IF (IO .NE. 0) THEN
      PRINT*,'*****************************************'
      PRINT*,'  OUTPUTS HAVE BEEN WRITTEN ON FILE #',IO
      PRINT*,'*****************************************'
      PRINT*
    ENDIF
    PRINT*,' '
    PRINT*,'      *************** END AURATEK **************'
    PRINT*,' '
  19 FORMAT(/,'OUTPUT FILE? TYPE FILE # OR 0  >:')
  29 FORMAT('DEPLOYMENT')
  39 FORMAT(A18,2(',',F8.2),',',F6.2,6(',',I1),1X,A16)
  49 FORMAT('END')
  59 FORMAT(' DO YOU WANT SYMBOLS & COLORS WRITTEN ON OUTPUT FILE?',
   +' Y OR N?',/' >  :')
  69 FORMAT( A18,3X,A1,3X,A1 )
  79 FORMAT(A40)
  89 FORMAT( A1)
    END
```

SUBROUTINE COORDS

FUNCTION: Subroutine COORDS allows the user to designate a point on the screen with the cursor, then prints out the coordinates of the cursor.

PARAMETERS: Alternate return

DISCUSSION: Subroutine COORDS allows the user to designate a point on the screen with the cursor, then prints out the coordinates of the cursor. This subroutine uses the cursor control device (i.e. mouse) input to locate the coordinates.

COORDS VARIABLE GLOSSARY:

Variable	Definition	Type	Units
ILINE	Holds x and y coordinates to print to the screen.	Char.	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: NONE

Subroutines calling COORDS: TARMOD WPNMOD

```
C*****************************COORDS*****************************************
      SUBROUTINE COORDS(*)
C
C  THE COORDS OPTION ALLOWS THE USER TO DESIGNATE A POINT ON
C  THE SCREEN WITH THE CURSORS, THEN PRINTS
C  OUT THE COORDINATES OF THE CURSORS.
C
      CHARACTER*40 ILINE
C
C  ACTIVATES MOUSE.
C
      CALL JLOCAT(1,1,1,b,X1,Y1)
C
C  CONVERTS FROM VIRTUAL COORDINATES TO WORLD COORDINATES.
C
      CALL JCONVW(X1,Y1,X,Y,Z)
C
C  PRINTS OUT COORDINATES FOUND FROM MOUSE INPUT.
C
      WRITE(ILINE,9) X,Y
    9 FORMAT( 2F9.1 )
      PRINT*,ILINE
      RETURN1
      END
```

SUBROUTINE DEPLOYA

FUNCTION: Subroutine DEPLOYA allows the user to deploy targets on the screen with the cursor.

PARAMETERS: NONE

DISCUSSION: Subroutine DEPLOYA allows the user to deploy targets on the screen with the cursor. The cursor control (i.e. mouse) input is used to deploy points. The user has the option to choose which symbol and color to use for that deployment point. This subroutine is called from the main menu in TARMOD.

DEPLOYA VARIABLE GLOSSARY:

Variable	Definition	Type	Units
ALTPOS	Holds alternate posture values.	Char.	N/A
DEFAULT	Flags if default postures are to be used.	Logical	N/A
EX	Difference between X maximum and X minimum values found in ZOOM.	Real	N/A
EY	Difference between Y maximum and Y minimum values found in ZOOM.	Real	N/A
FXM	Minimum value of FXM and XX found in ZOOM.	Real	Meters
FXP	Minimum value of FXP and XX found in ZOOM.	Real	Meters
FYM	Minimum value of FYM and YY found in ZOOM.	Real	Meters
FYP	Minimum value of FYP and YY found in ZOOM.	Real	Meters
ICH	Flags to clear previous weapon storage.	Char.	N/A
ICLR	Holds number of selected color.	Char.	N/A
ICOLOR	Sets PVI color value for symbol written to the output file.	Integer	N/A
ILINE	Holds characters of input.	Char.	N/A
IPAR	Array of kill criteria & initial postures.	Integer	N/A
INITDEP	Flags the first time a deployment	Logical	N/A

	option is used.		
ISYM	Sets PVI symbol value written to the output file.	Integer	N/A
ITEMP	Holds the rest of ILINE.	Char.	N/A
N1	Counts the number of commas in ILINE.	Integer	N/A
NAME	Asset names from input file.	Char.	N/A
NDEPL	Number of deployment points.	Integer	N/A
PAR	Array holding value of PRT.		
PRT	Real of integer IPRT.	Real	N/A
REDRAW	Flag to see if screen needs to be redrawn.	Logical	N/A
TXM	Minimum value of XZ(1) and XZ(2).	Real	Meters
TXP	Maximum value of XZ(1) and XZ(2).	Real	Meters
TYM	Minimum value of YZ(1) and YZ(2).	Real	Meters
TYP	Maximum value of YZ(1) and YZ(2).	Real	Meters
XTAR	Array holding X and Y coordinates.	Real	Meters

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: AMIN1, AMAX1, FLOAT and INDEX

Subroutines calling DEPLOYA: TARMOD

```
C*****************************DEPLOY*****************************************
      SUBROUTINE DEPLOYA
C
C  THE DEPLOY OPTION ALLOWS THE USER TO DEPLOY TARGETS ON THE
C  SCREEN WITH THE CURSORS.  THE USER MAY OPT TO USE THE DEFAULTS
C  FOR POSTURES AND KILL CRITERIA (ALL 1'S EXCEPT MOPP = 0).  THE
C  USER GIVES A NAME TO THE NEW TARGET AND CAN DESIGNATE A SYMBOL
C  AND COLOR FOR THIS TARGET.
C
C
      COMMON/WIDTH/ EX, EY
      COMMON/FLAG/ REDRAW
      COMMON/LOGS/ NEWSCR, DEFAULT, DUMMY, ASSETS, INITDEP
      COMMON/PAD/TXM,TXP,TYM,TYP, LEFT, FACTOR, TOP,FXM,FXP,FYM,FYP
      COMMON/NAMES/NAME(NDEPMX),ISYM(NDEPMX),ICOLOR(NDEPMX),
     + ALTPOS(NDEPMX), COMM(NDEPMX)
      COMMON/DEPLOY/ NDEPL, XTAR(NDEPMX,2), IPAR(NDEPMX,7)
      CHARACTER*18 NAME,ITEMP,ILINE*40,ID,ALTPOS*40,COMM*40
      CHARACTER*1 ISYM, ICHAR, ICH, ICLR, ICOLOR
      PARAMETER ( NDEPMX = 702 )
      INTEGER SYM, COL
      LOGICAL DEFAULT, INITDEP, REDRAW, NEWSCR, DUMMY, ASSETS
      DIMENSION PAR(NDEPMX,7)
      DIMENSION IPRT(7)
      EQUIVALENCE ( PAR,IPAR )

C
C  DEPLOY A TARGET AT CURSOR.
C
C  INITDEP IS A LOGICAL VARIABLE USED TO FLAG THE FIRST TIME A
C  DEPLOYMENT OPTION HAS BEEN USED.  THIS WAY THE USER IS ONLY
C  ASKED ONCE WHETHER THE DEFAULTS SHOULD BE USED.
C
      IF( EX .LT. 1.0E-6 .AND. EY .LT. 1.0E-6 )  THEN
        PRINT*,'MUST SET SCREEN FIRST'
      ELSE
        IF (INITDEP) THEN
          PRINT*,'USE DEFAULTS FOR KC AND POSTURES? >'
          READ(*,9)ICHAR
    9     FORMAT(A1)
          IF (ICHAR .EQ. 'Y') THEN
            DEFAULT = .TRUE.
          ENDIF
          INITDEP = .FALSE.
        ENDIF
        PRINT*,'SYMBOL        COLOR'
        PRINT*,'----------------------'
        PRINT*,'1 -  .     1 - Red'
        PRINT*,'2 -  +     2 - Green'
        PRINT*,'3 -  *     3 - Yellow'
        PRINT*,'4 -  O     4 - Blue'
        PRINT*,'5 -  X     5 - Purple'
        PRINT*,'           6 - Cyan'
```

```
      PRINT*,'         7 - White'
      PRINT*,' '
      PRINT*,'Enter Marker number'
      READ(*,19)SYM
      PRINT*,'Enter Marker color'
      READ(*,19)COL
   19  format(I1)
C
C  OPENS PVI GRAPHICS WINDOW.
C
      CALL JOPEN
C
C  PVI ROUTINE ASKS FOR INPUT FROM CURSOR CONTROLLER (MOUSE).
C
      CALL JLOCAT(1,1,1,b,XZ,YZ)
C
C  PVI ROUTINE CONVERTS FROM VIRTUAL COORDINATES TO WORLD
C  COORDINATES.
C
      CALL JCONVW(XZ,YZ,XX,YY,Z)
C
C  PVI ROUTINES THAT SETS MARKER (SYMBOL) AND COLOR TO THOSE
C  SELECTED ABOVE.
C
      CALL JCOLOR(COL)
      CALL JCMARK(SYM)
C
C  PVI ROUTINE PLACES MARKER AT POSITION XX, YY.
C
      CALL JMARK(XX,YY)
C
C  PVI ROUTINE CLOSES GRAPHICS WINDOW.
C
      CALL JCLOSE
      ILINE = ' '
      PRINT*,'NAME(,SYM,COLOR) > '
      READ(*,29)ILINE
   29  FORMAT(A40)
C
C  PULLS OUT NAME SYMBOL AND COLOR FROM THE ABOVE STRING ILINE.
C
      N1 = INDEX(ILINE,',')
      IF( N1 .LE. 0 )  GO TO 1338
      ID = ILINE(:N1-1)
      ICH = ILINE(N1+1:N1+1)
      ITEMP = ILINE(N1+2:)
      N1 = INDEX(ITEMP,',')
      IF (N1 .LE. 0) GO TO 1340
C
C  ICLR, IF DESIGNATED, IS A ONE CHARACTER VARIABLE USED TO
C  INDICATE THE COLOR OF THE TARGET MARKER.
C
      ICLR = ITEMP(N1+1:N1+1)
```

```
      GO TO 1340
 1338   ID = ILINE
      ICH = ' '
 1340   CONTINUE
      ILINE = ' '
      IF (.NOT. (DEFAULT)) THEN
        PRINT*,'NHERE,CRITERIA,POSTURES >'
        READ(*,39)ILINE
  39     FORMAT(a40)
C
C  PULLS OUT POSTURE VALUES AND KILL CRITERIA FROM THE ABOVE
C  STRING.
C
        DO 10  J = 1, 6
          N1 = INDEX(ILINE,',')
          IF ( N1 .LE. 0 ) THEN
          print*,'NEED 7 ENTRIES IN 36 COLUMNS'
          GO TO 1340
          ENDIF
          ITEMP = ILINE(:N1-1)
          N0 = INDEX(ITEMP,'.')
          IF( N0 .LE. 0 )  THEN
            READ(ITEMP,49,ERR=7734) IPRT(J)
  49          FORMAT( BN,I6 )
            IF( J .EQ. 1 )  PRT = FLOAT(IPRT(1))
          ELSE IF( J .EQ. 1 )  THEN
            READ(ITEMP,59,ERR=7734) PRT
  59          FORMAT( F18.3 )
          ELSE
            GO TO 1340
          ENDIF
          ILINE = ILINE(N1+1:)
  10      CONTINUE
        READ(ILINE,1359,ERR=7734) IPRT(7)
      ELSE
C
C  ASKS FOR THE NUMBER OF ASSETS DEPLOYED AT THIS POINT.
C
        PRINT*,'number @ this point >( integer only )'
        READ(*,69)ILINE
  69     FORMAT(A40)
C
C  CONVERTS ABOVE STRING TO INTEGER VALUE.
C
        N0 = INDEX(ILINE,'.')
        IF (N0 .LE. 0) THEN
          READ(ILINE,49,ERR=7734) IPRT(1)
          PRT = FLOAT(IPRT(1))
        ELSE IF (N0 .EQ. 1) THEN
          READ(ILINE,49,ERR=7734) PRT
        ELSE
          GO TO 1340
        ENDIF
```

```
        DO 20 I = 2,6
 20       IPRT(I) = 1
        IPRT(7) = 0
      ENDIF
      NDEPL = NDEPL+1
C
C  CHECKS TO SEE IF THE NUMBER DEPLOYED IS GREATER THAN MAXIMUM
C  NUMBER OF DEPLOYMENT POINTS.
C
      IF( NDEPL .GT. NDEPMX )  THEN
1328     WRITE(ILINE,79)
 79     FORMAT('TOO MANY DEPLOYMENTS!')
        print*,ILINE
        NDEPL = NDEPL-1
      ELSE
C
C  IF NOT TOO MANY DEPLOYMENT POINTS IT SETS VALUES TO THE NEWLY
C  SELECTED VALUES.
C
        NAME(NDEPL) = ID
        ISYM(NDEPL) = ICH
        ICOLOR(NDEPL) = ICLR
        XTAR(NDEPL,1)=XX
        XTAR(NDEPL,2)=YY
        PAR(NDEPL,1) = PRT
        DO 30 J = 2, 7
          IPAR(NDEPL,J) = IPRT(J)
 30     CONTINUE
        FXM = AMIN1(FXM,XX)
        FXP = AMAX1(FXP,XX)
        FYM = AMIN1(FYM,YY)
        FYP = AMAX1(FYP,YY)
      ENDIF
    ENDIF
    GO TO 9999
7734 CONTINUE
    PRINT*,'INPUT ERROR!  TRY AGAIN!'
9999 RETURN
    END
```

SUBROUTINE DISTANC

FUNCTION: Subroutine DISTANC allows the user to use the cursor to designate two points on the screen. It then displays the distance between those points.

PARAMETERS: NONE

DISCUSSION: Subroutine DISTANC allows the user to use the cursor to designate two points on the screen. It then displays the distance between those points. This routine uses the cursor control device (ie. mouse) to designate the points needed to calculate the distance.

DISTANC VARIABLE GLOSSARY:

Variable	Definition	Type	Units
D	Distance between the two points.	Real	Meters

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: LOCATE

System Library functions: NONE

Subroutines calling DISTANC: TARMOD WPNMOD

```
C*******************************DISTANC************************************
      SUBROUTINE DISTANC
C
C  THE DISTANCE OPTION ALLOWS THE USER TO USE THE CURSORS TO
C  DESIGNATE TWO POINTS ON THE SCREEN.  THIS ROUTINE THEN DISPLAYS
C  THE USER THE DISTANCE BETWEEN THOSE POINTS.
C
      CHARACTER ID*10
C
C  PVI ROUTINES TO OPEN GRAPHICS, SET COLOR AND SET MARKER
C  (SYMBOL).
C
      CALL JOPEN
      CALL JCOLOR(1)
      CALL JCMARK(2)
C
C     PROMPT TO TELL USER TO DESIGNATE POINTS.
C
      PRINT*,'DESIGNATE POINTS'
C
C  PVI ROUTINE TURNS ON CURSOR CONTROL DEVICE (MOUSE) TO FIND POINT 1.
C
       CALL JLOCAT(1,1,1,B,XZ,YZ)
C
C  PVI ROUTINE CONVERTS FROM VIRTUAL TO WORLD COORDINATES.
C
       CALL JCONVW(XZ,YZ,X1,Y1,Z)
C
C  PVI ROUTINE PLACES MARKER (SET TO SYMBOL OF THE VALUE IN JCMARK)
C  AT X1,Y1.
C
       CALL JMARK(X1,Y1)
C
C  PVI ROUTINE TURNS ON CURSOR CONTROL DEVICE (MOUSE) TO FIND POINT 2.
C
       CALL JLOCAT(1,1,1,b,XJ,YJ)
       CALL JCONVW(XJ,YJ,X2,Y2,Z)
       CALL JMARK(X2,Y2)
C
C  CALCULATES DISTANCE.
C
     D = SQRT( (X1-X2)**2 + (Y1-Y2)**2 )
     WRITE(ID,9) D
   9 FORMAT( F10.3 )
     PRINT*," "
     PRINT*,ID
     RETURN
     END
```

SUBROUTINE EMPLOY

FUNCTION: Subroutine EMPLOY is used to graphically display weapon burst locations on the screen.

PARAMETERS: NONE

DISCUSSION: Subroutine EMPLOY is used to graphically display weapon burst locations on the screen. For a chemical weapon laydown an elliptical pattern is drawn with each chemical burst point. This pattern is not to be used for weapon effectiveness. For a conventional weapon laydown an asterisk with a circle is drawn for each burst point. For a nuclear weapon laydown an asterisk is drawn for each burst point. Subroutine REFRESH is called to draw the screen after new dimensions are set because of the weapon deployment.

EMPLOY VARIABLE GLOSSARY:

Variable	Definition	Type	Units
ISTART	Value to start weapon number.	Integer	N/A
LCENT	Flag to mark weapon centers.	Logical	N/A
KF	Set to value of IWIN.	Integer	N/A
KRDS	Number of rounds from weapon file.	Integer	N/A
KREP	Number of replications from file.	Integer	N/A
KWIN	Set to the value of WIND.	Real	N/A
KWPN	Set to the value of MWPN.	Integer	N/A
NF	Counts the number of weapon files used.	Integer	N/A
NFIL	Holds value of IWIN.	Integer	N/A
NOWPN	Set to the value of NWTEMP.	Integer	N/A
NPTS	Number of points to draw the ellipse.	Integer	N/A
NWTOTL	Total number of weapons read in.	Integer	N/A
PASS1	Flags for all replications of weapon.	Logical	N/A
REDRAW	Flags if screen needs to be redrawn.	Logical	N/A
REPLS	Flags to ask employ question.	Logical	N/A
SCRMAX	Maximum length with weapons deployed.	Real	N/A
TXM	Set to the value of TXXM.	Real	N/A
TXP	Set to the value of TXXP.	Real	N/A
TXXM	Maximum screen coordinate in X direction.	Real	N/A

TXXP	Minimum screen coordinate in X direction.	Real	N/A
TYM	Set to the value of TYYM.	Real	N/A
TYP	Set to the value of TXXM.	Real	N/A
TYYM	Maximum screen coordinate in Y direction.	Real	N/A
TYYP	Minimum screen coordinate in Y direction.	Real	N/A
USRPAT	Flags if USRPAT has been called before.	Logical	N/A
UXMAX	Maximum X of weapon pattern.	Real	N/A
UXMIN	Minimum X of weapon pattern.	Real	N/A
UYMAX	Maximum Y of weapon pattern.	Real	N/A
UYMIN	Minimum Y of weapon pattern.	Real	N/A
WXM	Set to the value of TXM.	Real	N/A
WXMAX	Set to the value of WXMX.	Real	N/A
WXMIN	Set to the value of WXMN.	Real	N/A
WXP	Set to the value of TXP.	Real	N/A
WYM	Set to the value of TYM.	Real	N/A
WYMAX	Set to the value of WYMX.	Real	N/A
WYMIN	Set to the value of WYMN.	Real	N/A
WYP	Set to the value of TYP.	Real	N/A
XDIAM	Half of length in X direction.	Real	N/A
XKA	X and Y coordinate to mark weapon.	Real	N/A
XPAT	X coordinate used to draw ellipse.	Real	N/A
YDIAM	Half of length in Y direction.	Real	N/A
YPAT	Y coordinate used to draw ellipse.	Real	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: PATTERN REFRESH

System Library functions: AMAX1, ABS

Subroutines calling EMPLOY: WPNMOD

```
C*******************************EMPLOY*****************************************
      SUBROUTINE EMPLOY
C
C     THE EMPLOY OPTION IS CALLED BY WPNMOD AND IS USED TO EMPLOY
C     INCOMING WEAPONS ON THE SCREEN.
C
      COMMON/REPS/ REPLS
      COMMON/WPNFILE/ WFILE(NFILES), NCLR(NFILES), NF
      COMMON/PTN/ USRPAT(NWPNS)
      COMMON/PTNDAT/UXMIN(NWPNS),UXMAX(NWPNS),UYMIN(NWPNS),UYMAX(NWPNS)
      COMMON/FLAG/ REDRAW
      COMMON/WDIM/ WXM, WXP, WYM, WYP
      COMMON/PAD/TXM,TXP,TYM,TYP, LEFT, FACTOR, TOP,FXM,FXP,FYM,FYP
      COMMON/POINTS/ XPAT(NPTS), YPAT(NPTS)
      COMMON/WPNDAT/KREP(NRDMAX),KWPN(NRDMAX),TIMK(NRDMAX),KF(NRDMAX),
     + XKA(NRDMAX,3),XKD(NRDMAX,3),KWIN(NRDMAX),KRDS(NFILES),NFIL(NWPNS)
     + ,NWEAPS(NFILES), IREP(NWPNS), NOWPN(NWPNS), WXMIN(NWPNS), NWPN,
     + WXMAX(NWPNS), WYMIN(NWPNS), WYMAX(NWPNS), WIND(NRDMAX),NWTOTL
      COMMON/WPNCHR/ IWPN(NWPNS), WPNAME(NWPNS)
      COMMON/SCRFLG/ INITEMP
      CHARACTER*18 IWPN, WPNAME, ILINE*60, ANS*1, ANS2*1
      PARAMETER ( NFILES = 10 )
      PARAMETER ( NPTS = 30 )
      PARAMETER ( NWPNS = 20 )
      PARAMETER ( NRDMAX = 1150 )
      LOGICAL INITEMP, REDRAW, USRPAT, REPLS, PASS1, LCENT
      PASS1 = .TRUE.
C
C  IF FIRST TIME, RESCALE SCREEN TO ENSURE NO PATTERNS ARE LOST.
C
      SCRMAX = 0.0
      DO 10 J = 1, NWTOTL
C
C  FINDS THE MAXIMUM LIMITS OF THE SCREEN BY COMPARING DEPLYOMENT
C  POINTS.
C
        DO 20 I = 1, KRDS(NF)
          IF ( NOWPN(J) .EQ. KWPN(I) .AND. NFIL(J) .EQ. KF(I)) THEN
            SCRMAX = AMAX1(SCRMAX,ABS(YMAX1) + ABS(XKA(I,2)))
            SCRMAX = AMAX1(SCRMAX,ABS(YMIN1) + ABS(XKA(I,2)))
            SCRMAX = AMAX1(SCRMAX,ABS(XMAX1) + ABS(XKA(I,1)))
            SCRMAX = AMAX1(SCRMAX,ABS(XMIN1) + ABS(XKA(I,1)))
          ENDIF
   20   CONTINUE
   10 CONTINUE
C
C  SET SCREEN WIDTH PARAMETERS AND REFRESH SCREEN.
C
      XDIAM = (TXP - TXM) / 2.0
      YDIAM = (TYP - TYM) / 2.0
      IF ((SCRMAX .GT. XDIAM) .OR. (SCRMAX .GT. YDIAM)) THEN
        PRINT*,'SOME WEAPON PATTERNS DO NOT FIT ON SCREEN'
        PRINT*,'RESCALE (Y OR N) >'
```

```
      READ(*,9)ANS
    9 FORMAT(A1)
C
C THIS SECTION RESCALES THE SCREEN TO FIT THE WEAPON PATTERN
C OVER THE DEPLOYMENT.
C
      IF (ANS .EQ. 'Y') THEN
        TXXM = TXM - SCRMAX * 1.001
        TXXP = TXP + SCRMAX * 1.001
        TYYM = TYM - SCRMAX * 1.001
        TYYP = TYP + SCRMAX * 1.001
        IF ((TXM .NE. WXM) .OR. (TXP .NE. WXP) .OR.
     +      (TYM .NE. WYM) .OR. (TYP .NE. WYP)) THEN
          TXM = TXXM
          TXP = TXXP
          TYM = TYYM
          TYP = TYYP
          WXM = TXM
          WXP = TXP
          WYM = TYM
          WYP = TYP
          WRITE(ILINE,19) TXM,TXP,TYM,TYP
   19      FORMAT(4F9.0)

          CALL REFRESH
          PRINT*, 'SCREEN RESCALED FOR WEAPON EMPLOYMENT!'
          PRINT*,' '
          PRINT*,'XMIN, XMAX, YMIN, YMAX = '
          PRINT*,ILINE
        ENDIF
      ENDIF
    ENDIF
C
C PVI ROUTINE TO OPEN GRAPHICS WINDOW.
C
      CALL JOPEN
    IF (REPLS) THEN
C
C CHOOSE MARK WEAPON CENTERS AND ALL REPLICATIONS.
C
      PRINT*,'EMPLOY ALL REPLICATIONS OR'
      PRINT*,'WALK THROUGH (A OR W)? >'
      READ(*,29)ANS
   29 FORMAT(A1)
    ENDIF
    PRINT*,'MARK CENTERS (Y OR N)? >'
    READ(*,39)ANS
   39 FORMAT(A1)
    IF( ANS .EQ. 'Y' ) THEN
      LCENT = .TRUE.
    ELSE
      LCENT = .FALSE.
    ENDIF
```

```
      NCOLOR = 2
      DO 30 K = 1, NF
        IF (K .EQ. 1) THEN
          ISTART = 1
        ELSE
          ISTART = KRDS(K-1) + 1
        ENDIF
C
C  PVI ROUTINE TO SET THE COLOR OF WEAPON PATTERN.
C
        CALL JCOLOR(1)
        DO 40 I = ISTART, KRDS(K)
          IF(XKA(I,1) .LT. TXM .OR. XKA(I,1) .GT. TXP ) GO TO 1420
          IF(XKA(I,2) .LT. TYM .OR. XKA(I,2) .GT. TYP )  GO TO 1420
          IF (PASS1) THEN
            IF (ANS .EQ. 'W') THEN
              IF (I .NE. 1) THEN
                IF (KREP(I) .NE. KREP(I-1)) THEN
                  NCOLOR = NCOLOR + 1
                  IF (NCOLOR .GT. 7) THEN
                  NCOLOR = 2
                  CALL JCOLOR(NCOLOR)
                  PRINT*,'CONTINUE WITH NEXT REPLICATION? >'
                  READ(*,49)ANS2
 49                FORMAT(A1)
                  IF (ANS2 .EQ. 'N') GO TO 1415
                  ENDIF
                ENDIF
              ENDIF
            ENDIF
          ENDIF
          DO 50 J = 1, NWTOTL
            IF( LCENT ) THEN
C
C  PVI ROUTINE TO DRAW THE WEAPON PATTERN.
C
              CALL JCMARK(3)
              CALL JMARK(XKA(I,1),XKA(I,2))
C
C SUBROUTINE PATTERN CALCULATES THE ELLIPTICAL WEAPON PATTERN.
C
              CALL PATTERN(I,XKA,XMAX1,XMIN1,YMAX1,YMIN1,KWIN(I))
              CALL JMOVE(XPAT,YPAT)
              CALL JPOLY(XPAT,YPAT,NPTS)
              ENDIF
            ENDIF
  50      CONTINUE
  40    CONTINUE
        CALL JCOLOR(7)
  30  CONTINUE
      REDRAW = .TRUE.
C
C  PVI ROUTINE TO CLOSE THE GRAPHICS WINDOW.
```

```
C
      CALL JCLOSE
      RETURN
      END
```

SUBROUTINE ENDTEK

FUNCTION: Subroutine ENDTEK turns off PVI graphics.

PARAMETERS: NONE

DISCUSSION: Subroutine ENDTEK turns off PVI graphics. This subroutine calls PVI subroutines to turn off device 1 (screen) and device 3 (hardcopy file), then terminate the DI-3000 session.

ENDTEK VARIABLE GLOSSARY:

Variable	Definition	Type	Units
---	---	---	---

NO SUBROUTINE VARIABLES.

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: NONE

Subroutines calling ENDTEK: TARMOD

```
C*******************************ENDTEK*****************************************
      SUBROUTINE ENDTEK
C
C     ENDTEK ENDS THE PROGRAM.
C     THIS TERMINATES THE PVI DI-3000 GRAPHICS.
C
      CALL JPAUSE(1)
C
C  CLOSES DEVICE 1.
C
      CALL JDEVOF(1)
      CALL JDEND(1)
C
C  CLOSES DEVICE 3.
C
      CALL JDEVOF(3)
      CALL JDEND(3)
C
C  ENDS PLOT.
C
      CALL JEND
      RETURN
      END
```

SUBROUTINE FULLSC

FUNCTION: Subroutine FULLSC returns the screen to its original dimensions displaying all targets.

PARAMETERS: NONE

DISCUSSION: Subroutine FULLSC returns the screen to its original dimensions displaying all targets. Subroutine REFRESH is called to draw the screen at the new dimensions.

FULLSC VARIABLE GLOSSARY:

Variable	Definition	Type	Units
EX	Difference between X maximum and X minimum values found in ZOOM.	Real	N/A
EY	Difference between Y maximum and Y minimum values found in ZOOM.	Real	N/A
FXM	Minimum value of FXM and XX found in ZOOM.	Real	Meters
FXP	Maximum value of FXP and XX found in ZOOM.	Real	Meters
FYM	Minimum value of FYM and YY found in ZOOM.	Real	Meters
FYP	Maximum value of FYP and YY found in ZOOM.	Real	Meters
REDRAW	Flag to see if screen needs to be redrawn.	Logical	N/A
TXM	Minimum value of XZ(1) and XZ(2) found in ZOOM.	Real	Meters
TXP	Maximum value of XZ(1) and XZ(2) found in ZOOM.	Real	Meters
TYM	Minimum value of YZ(1) and YZ(2) found in ZOOM.	Real	Meters
TYP	Maximum value of YZ(1) and YZ(2) found in ZOOM.	Real	Meters

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: REFRESH

System Library functions: NONE

Subroutines calling FULLSC: TARMOD WPNMOD

```
C*********************************FULLSC************************************
      SUBROUTINE FULLSC
C
C  FULL SCALE RETURNS THE SCREEN TO ITS ORIGINAL DIMENSIONS,
C  DISPLAYING ALL TARGETS.
C

      COMMON/FLAG/ REDRAW
      COMMON/PAD/TXM,TXP,TYM,TYP, LEFT, FACTOR, TOP,FXM,FXP,FYM,FYP
      LOGICAL REDRAW

      TXM = FXM
      TXP = FXP
      TYM = FYM
      TYP = FYP
      EX = TXP-TXM
      EY = TYP-TYM
C
C  SUBROUTINE REFRESH IS CALLED TO DRAW THE SCREEN AT THE NEW
C  DIMENSIONS.
C
      CALL REFRESH
      REDRAW = .TRUE.
      RETURN
      END
```

SUBROUTINE GRID

FUNCTION: Subroutine GRID draws a grid on the current screen and prints the tic intervals.

PARAMETERS: CALL GRID (TXMIN= Minimum X coordinate of deployment, TXMAX= Maximum X coordinate of deployment, TYMIN= Minimum Y coordinate of deployment, YMAX= Maximum Y coordinate of deployment)

DISCUSSION: Subroutine GRID draws a grid on the current screen and prints the tic intervals.

GRID VARIABLE GLOSSARY:

Variable	Definition	Type	Units
AXIS	Greater value of XLEN or YLEN.	Real	Meters
TIC	One tenth the value of AXIS.	Real	Meters
TMOVE	Set to the value of TIC.	Real	Meters
XLEN	Length in X direction.	Real	Meters
YLEN	Length in Y direction.	Real	Meters

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: AMAX1

Subroutines calling GRID: TARMOD WPNMOD

```
C*****************************GRID*****************************************
      SUBROUTINE GRID(TXMIN,TXMAX,TYMIN,TYMAX)
C
C  THE GRID OPTION DRAWS A GRID ON THE CURRENT SCREEN AND PRINTS
C  THE TIC INTERVAL FOR THE USER.
C
C  THE TIC INTERVAL IS KEPT THE SAME FOR BOTH X AND Y AXES.
C
      XLEN = ABS(TXMAX - TXMIN)
      YLEN = ABS(TYMAX - TYMIN)
      IF (XLEN .NE. YLEN) THEN
        AXIS = AMAX1(XLEN,YLEN)
      ELSE
        AXIS = XLEN
      ENDIF
      TIC = AXIS/10.0
      TMOVE = TIC
C
C  DRAW GRID AXES.
C
      CALL JOPEN
      CALL JCOLOR(3)
      CALL JMOVE(TXMAX,TYMIN)
      CALL JDRAW(TXMIN,TYMIN)
      CALL JDRAW(TXMIN,TYMAX)
      CALL JMOVE(TXMIN,TYMIN)
C
C  DRAW TIC MARKS ON AXES.
C
      DO 10 I = 1,40
        IF ((TXMIN + TMOVE) .LE. TXMAX) THEN
          CALL JDRAW(TXMIN + TMOVE,TYMIN)
          CALL JDRAW(TXMIN + TMOVE,TYMIN + YLEN*0.02)
          CALL JMOVE(TXMIN + TMOVE,TYMIN)
          TMOVE = TMOVE + TIC
        ENDIF
   10 CONTINUE
      TMOVE = TIC
      CALL JMOVE(TXMIN,TYMIN)
      DO 20 I = 1,40
        IF ((TYMIN + TMOVE) .LE. TYMAX) THEN
          CALL JDRAW(TXMIN,TYMIN + TMOVE)
          CALL JDRAW(TXMIN + XLEN*0.02,TYMIN + TMOVE)
          CALL JMOVE(TXMIN,TYMIN + TMOVE)
          TMOVE = TMOVE + TIC
        ENDIF
   20 CONTINUE
      PRINT*,'TIC INTERVAL IS',TIC
      CALL JCLOSE
      RETURN
      END
```

SUBROUTINE HEADER

FUNCTION: Subroutine HEADER enables the user to put a header on various portions of the screen in different sizes.

PARAMETERS: NONE

DISCUSSION: Subroutine HEADER enables the user to put a header on various portions of the screen in different sizes. The user has the option to have the text size small, medium or large. The function LSTCHR is used to find the value of the last non-blank character in the string TITLE2.

HEADER VARIABLE GLOSSARY:

Variable	Definition	Type	Units
YSIZE	Used to scale letters to layout.	Real	N/A
SET	Used to set different size letters.	Real	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: LSTCHR

Subroutines calling HEADER: TARMOD WPNMOD

```
C*************************HEADER*********************************************
      SUBROUTINE HEADER
C
C  HEADER ENABLES THE USER TO PUT A HEADER ON VARIOUS PORTIONS
C  OF THE SCREEN (FOR USE IN REPORTS, VIEWGRAPHS, ETC.)
C
      COMMON/PAD/TXM,TXP,TYM,TYP, LEFT, FACTOR, TOP,FXM,FXP,FYM,FYP
      CHARACTER*80 IHEAD, IANS*1, TITLE2*41
C
C  PVI ROUTINE  TO OPEN SEGMENT FOR HEADER.
C
      CALL JOPEN
      CALL JGAP(0.0)
      CALL JFONT(7)
      YSIZE = (FYP - FYM)*.5
      CALL JJUST(2,2)
C
C PROMPT THE USER FOR THE TEXT SIZE.
C
      PRINT*,'Enter text size (S,M,L)'
      READ(*,9),IANS
    9 FORMAT(A1)
      IF (IANS .EQ. 'S') THEN
        SET = 20.0
      ELSE IF (IANS .EQ. 'M') THEN
        SET = 8.0
      ELSE IF (IANS .EQ. 'L') THEN
        SET = 4.0
      ENDIF
      CALL JSIZE(YSIZE/SET ,YSIZE/SET )
C
C PRINT COLOR OPTIONS FOR THE HEADER.
C
      PRINT*,'COLORS AVAILABLE FOR HEADING ARE:'
      PRINT*,'RED, WHITE, BLUE, YELLOW, GREEN, CYAN & VIOLET.'
   10 PRINT*,' '
      PRINT*,'ENTER HEADING COLOR DESIRED >'
      READ(*,19),IANS
   19 FORMAT(A1)
      IF (IANS .EQ. 'R') THEN
        CALL JCOLOR(1)
      ELSE IF (IANS .EQ. 'W') THEN
        CALL JCOLOR(7)
      ELSE IF (IANS .EQ. 'B') THEN
        CALL JCOLOR(4)
      ELSE IF (IANS .EQ. 'Y') THEN
        CALL JCOLOR(3)
      ELSE IF (IANS .EQ. 'G') THEN
        CALL JCOLOR(2)
      ELSE IF (IANS .EQ. 'C') THEN
        CALL JCOLOR(6)
      ELSE IF (IANS .EQ. 'V') THEN
        CALL JCOLOR(5)
```

```
      ELSE
        PRINT*,'INPUT ERROR!  TRY AGAIN!'
        GO TO 10
      ENDIF
C
C  ENABLE CROSSHAIRS TO PLACE LABEL.
C
      PRINT*,'INDICATE CENTER OF LABEL WITH CURSOR'
        CALL JLOCAT(1,1,1,B,XZ,YZ)
        CALL JCONVW(XZ,YZ,X,Y,Z)
      PRINT*,' '
      PRINT*,'ENTER HEADER >'
      READ(*,29)IHEAD
 29   FORMAT(A80)
      WRITE (TITLE2,39)IHEAD
 39   FORMAT (A41)
C
C  PRINTS HEADER.
C
C  LSTCHR RETURNS THE VALUE OF THE LAST NON-BLANK CHARACTER IN
C  THE STRING.
C
      LENG = LSTCHR ( TITLE2 )
      CALL JMOVE(X,Y)
      CALL JHSTRG(TITLE2(1:LENG))
      CALL JCOLOR(7)
      CALL JCLOSE
      RETURN
      END
```

SUBROUTINE HELP

FUNCTION: Subroutine HELP prints out an explanation of each option.

PARAMETERS: CALL HELP(CMD= String that holds first three characters of a command)

DISCUSSION: Subroutine HELP prints out an explantation of each option. The user inputs the first three characters of the command and a short description is printed.

HELP VARIABLE GLOSSARY:

Variable	Definition	Type	Units
CMD	Holds first three characters of command.	Char.	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: NONE

Subroutines calling HELP: TARMOD WPNMOD

```
C*****************************HELP*****************************************
      SUBROUTINE HELP(CMD)
C
C  THE HELP OPTION SIMPLY PRINTS OUT AN EXPLANATION OF EACH OPTION
C  AVAILABLE IN AURATEK.
C
      CHARACTER*3 CMD
      IF (CMD .EQ. 'BUL') THEN
        PRINT*,'BULK  -  ENABLES YOU TO MOVE, REMOVE,     '
        PRINT*,'         DUPLICATE OR LIST GROUPS OF TARGETS.  '
      ELSE IF (CMD .EQ. 'COO') THEN
        PRINT*,'COORDS - ENABLES YOU TO FIND OUT THE X,Y      '
        PRINT*,'         COORDINATES AT THE CURSOR LOCATION.  '
      ELSE IF (CMD .EQ. 'DEP') THEN
        PRINT*,'DEPLOY - ENABLES YOU TO INDICATE A      '
        PRINT*,'         LOCATION ON THE SCREEN AND DEPLOY A  '
        PRINT*,'         TARGET AT THAT POINT.  DEFAULT VALUES '
        PRINT*,'         CAN BE USED FOR KILL CRITERIA AND    '
        PRINT*,'         POSTURES, IF DESIRED.                '
      ELSE IF (CMD .EQ. 'DIS') THEN
        PRINT*,'DISTANCE - ENABLES YOU TO DETERMINE'
        PRINT*,'           THE DISTANCE BETWEEN ANY TWO POINTS '
        PRINT*,'           ON THE SCREEN.                      '
      ELSE IF (CMD .EQ. 'END') THEN
        PRINT*,'END - CAUSES THE PROGRAM TO END.  YOU MAY  '
        PRINT*,'      DESIGNATE WHAT FILE IS TO BE USED FOR THE'
        PRINT*,'      OUTPUT OR ENTER "0" TO INDICATE NO OUTPUT'
        PRINT*,'      FILE IS DESIRED.                    '
      ELSE IF (CMD .EQ. 'FUL') THEN
        PRINT*,'FULL SCALE - RESTORES THE SCREEN TO '
        PRINT*,'             ITS INITIAL DIMENSIONS.        '
      ELSE IF (CMD .EQ. 'GRI') THEN
        PRINT*,'GRID - DRAWS AXES ON THE BOTTOM AND LEFT SIDE  '
        PRINT*,'       OF THE SCREEN AND PRINTS OUT THE TIC    '
        PRINT*,'       INTERVAL.                               '
      ELSE IF (CMD .EQ. 'INP') THEN
        PRINT*,'INPUT - READS IN WEAPON DATA FROM TAPE SPECI-  '
        PRINT*,'        FIED BY THE USER.  ASKS THE USER WHAT  '
        PRINT*,'        DATA TO DISPLAY.  TAPE MUST BE A DUMP9 '
        PRINT*,'        TAPE FROM AN AURA RUN.                 '
        PRINT*,'                                               '
        PRINT*,'        INPUT ALSO RESCALES SCREEN TO ENSURE   '
        PRINT*,'        THAT NONE OF THE WEAPON PATTERNS       '
        PRINT*,'        OVERLAP THE SCREEN.                    '
      ELSE IF (CMD .EQ. 'BAC') THEN
        PRINT*,'BACK - "BACK" OR "END" WILL RETURN THE USER TO '
        PRINT*,'       THE NORMAL MODE.                        '
      ELSE IF (CMD .EQ. 'EMP') THEN
        PRINT*,'EMPLOY - EMPLOYS WEAPON PATTERNS ON SCREEN     '
      ELSE IF (CMD .EQ. 'HEL') THEN
        PRINT*,'HELP - PROVIDES A BRIEF DESCRIPTION OF     '
        PRINT*,'       EACH COMMAND.  FORMAT FOR HELP IS:      '
        PRINT*,'         HELP "COMMAND NAME"                   '
```

```
ELSE IF (CMD .EQ. 'IDE') THEN
  PRINT*,'IDENTIFY - IDENTIFIES THE TARGET AT THE       '
  PRINT*,'           LOCATION SPECIFIED BY THE CURSORS. '
  PRINT*,' '
  PRINT*,'           IF IN WEAPON MODE, IDENTIFIES WPN  '
  PRINT*,'           AT CURSOR.                         '
ELSE IF (CMD .EQ. 'LIS') THEN
  PRINT*,'LIST - LISTS ALL WEAPON #S AND NAMES.         '
ELSE IF (CMD .EQ. 'LOC') THEN
  PRINT*,'LOCATE - LOCATES A GIVEN X,Y COORDINATE ON THE '
  PRINT*,'         SCREEN AND PLACES A "+" THERE.       '
ELSE IF (CMD .EQ. 'MEN') THEN
  PRINT*,'MENU - PRINTS THE LIST OF AVAILABLE COMMANDS. '
ELSE IF (CMD .EQ. 'MOV') THEN
  PRINT*,'MOVE (OR REMOVE) - MOVES OR REMOVES THE       '
  PRINT*,'                   INDICATED TARGET.          '
ELSE IF (CMD .EQ. 'PRE') THEN
  PRINT*,'PREVIOUS - RESTORES PRECEDING SCREEN.         '
ELSE IF (CMD .EQ. 'REF') THEN
  PRINT*,'REFRESH - REDRAWS CURRENT SCREEN.             '
ELSE IF (CMD .EQ. 'RES') THEN
  PRINT*,'RESCALE - REMOVES DISTORTION OF SCREEN.       '
ELSE IF (CMD .EQ. 'SCR') THEN
  PRINT*,'SCREEN - DISPLAYS THE CURRENT MINIMUM AND     '
  PRINT*,'         MAXIMUM X AND Y VALUES FOR THE SCREEN.'
ELSE IF (CMD .EQ. 'SET') THEN
  PRINT*,'SET SCREEN - ENABLES YOU TO SET THE MINIMUM   '
  PRINT*,'             & MAXIMUM VALUES FOR THE X AND Y '
  PRINT*,'             SCREEN DIMENSIONS.               '
ELSE IF (CMD .EQ. 'SYM') THEN
  PRINT*,'SYMBOL - ENABLES YOU TO DESIGNATE SYMBOLS FOR '
  PRINT*,'         TARGETS.                             '
ELSE IF (CMD .EQ. 'WEA') THEN
  PRINT*,'WEAPON - ENABLES YOU TO ENTER WEAPON MODE.    '
ELSE IF (CMD .EQ. 'ZOO') THEN
  PRINT*,'ZOOM - EXPANDS A PART OF THE CURRENT SCREEN.  '
ELSE
  PRINT*,' ******  INPUT ERROR!  TRY HELP AGAIN!! ******  '
ENDIF
PRINT*,' '
RETURN
END
```

SUBROUTINE IDENTFY

FUNCTION: Subroutine IDENTFY allows the user to identify the target at the cursor.

PARAMETERS: CALL IDENTFY(NMV= Maximum number of assets at a point that program can identify, IFLG= flags to count the number of assets at a point.)

DISCUSSION: Subroutine IDENTFY allows the user to identify the target at the cursor. This subroutine will give the user kill criteria and posture for the selected asset or weapon information if a weapon is selected.

IDENTFY VARIABLE GLOSSARY:

Variable	Definition	Type	Units
COMAND	Holds value of COMMAND from TARMOD.	Char.	N/A
ID	Holds value of NAME.	Char.	N/A
IPAR	Holds array of kill criteria & initial postures.	Char.	N/A
ISYM	Holds PVI symbol character used to see if asset is visible.	Char.	N/A
ITMP	Set to the number of deployment points.	Integer	N/A
IVIS	Character denoting invisible point.	Char.	N/A
KF	Number assigned to weapon file.	Integer	N/A
KREP	Holds value for the number of replications.	Integer	N/A
KRDS	Set to the number of rounds in weapon file.	Integer	N/A
KWIN	Set to the wind value in weapon file.	Real	N/A
KWPN	Set to the weapon number from weapon file.	Integer	N/A
NDEPL	Number of deployment points.	Integer	N/A
NF	Number of weapon files used.	Integer	N/A
NFIL	Number assigned to weapon file.	Integer	N/A
NOWPN	Number assigned to weapon in weapon file.	Integer	N/A
NVIS	Count of how many symbols are visible.	Integer	N/A
NWTOTL	Total number of weapons in weapon file.	Integer	N/A
TIMK	Time read in from weapon input file.	Real	N/A

TXM	Min. value of XZ(1) and XZ(2) found in ZOOM	Real	Meters
TXP	Max. value of XZ(1) and XZ(2) found in ZOOM	Real	Meters
TYM	Min. value of YZ(1) and YZ(2) found in ZOOM	Real	Meters
TYP	Max. value of YZ(1) and YZ(2) found in ZOOM	Real	Meters
WPNAME	Weapon name from weapon input file.	Char.	N/A
XKA	X, Y & Z coordinates for function point of weapon.	Real	N/A
XKD	X, Y & Z coordinates for aim point of weapon.	Real	N/A
XTAR	Array of X & Y coordinates of asset deployment.	Real	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: ABS and INDEX

Subroutines calling IDENTFY: MOVSUB TARMOD WPNMOD

```
C*************************IDENTFY*****************************************
      SUBROUTINE IDENTFY(NMV,IFLG)
C
C  THE IDENTIFY OPTION ALLOWS THE USER TO IDENTIFY THE TARGET
C  (OR IF IN WEAPON MODE, THE WEAPON) AT THE CURSORS.
C
      COMMON/WPNFILE/ WFILE(NFILES), NCLR(NFILES), NF
      COMMON/COMAND/COMAND
      COMMON/TEMPS/ ITMP
      COMMON/CHRTMP/ ID
      COMMON/FLAG/ REDRAW
      COMMON/WPNDAT/KREP(NRDMAX),KWPN(NRDMAX),TIMK(NRDMAX),KF(NRDMAX),
     + XKA(NRDMAX,3),XKD(NRDMAX,3),KWIN(NRDMAX),KRDS(NFILES),NFIL(NWPNS)
     + ,NWEAPS(NFILES), IREP(NWPNS), NOWPN(NWPNS), WXMIN(NWPNS), NWPN,
     + WXMAX(NWPNS), WYMIN(NWPNS), WYMAX(NWPNS), WIND(NRDMAX),NWTOTL
      COMMON/WPNCHR/ IWPN(NWPNS), WPNAME(NWPNS)
      COMMON/VISBL/ IVIS(64)
      COMMON/VISINT/ NVIS
      COMMON/DEPLOY/ NDEPL, XTAR(NDEPMX,2), IPAR(NDEPMX,7)
      COMMON/PAD/TXM,TXP,TYM,TYP, LEFT, FACTOR, TOP,FXM,FXP,FYM,FYP
      COMMON/NAMES/NAME(NDEPMX),ISYM(NDEPMX),ICOLOR(NDEPMX),
     + ALTPOS(NDEPMX), COMM(NDEPMX)
      CHARACTER*18 NAME,ITEMP,ILINE*80,ID,NUMNUT, WPNAME*18, IWPN*18,
     + ALTPOS*40, COMM*40
      CHARACTER*1 ISYM, IVIS, IDSYM*5, ICOLOR, I,COMAND*18
      PARAMETER ( NRDMAX = 1150 )
      PARAMETER ( NFILES = 10 )
      PARAMETER ( NWPNS = 20 )
      PARAMETER ( NDEPMX = 702 )
      LOGICAL REDRAW
      DIMENSION  ITMP(100)
      DIMENSION PAR(NDEPMX,7)
      EQUIVALENCE ( PAR,IPAR )
      NMV = 0
C
C  RESOLUTION IS BASED ON THE CHARACTER WIDTH AND HEIGHT AS COMPARED
C  TO THE SIZE OF THE DISPLAY (ON A TEKTRONIX 4113).  TEKTRONIX 4113
C  VALUES WERE USED BECAUSE TEKTRONIX 4105 VALUES WERE NOT AVAILABLE
C  IN THE MOST RECENT IGL MANUAL.
C
      XDX = (TXP-TXM)*0.0126/2.
      YDY = (TYP-TYM)*0.031/2.
C
C  THE ABOVE ASSUME 1% FULL SCREEN RESOLUTION IN CURSOR POSITION.
C
C  PVI ROUTINE TO OPEN GRAPHICS WINDOW AND START CURSOR CONTROL
C  DEVICE.
C
        CALL JOPEN
        CALL JLOCAT(1,1,1,b,XZ,YZ)
        CALL JCONVW(XZ,YZ,X,Y,Z)
C
C  PVI SUBROUTINES TO SET COLOR & SYMBOL TO INDICATE ASSET BEING
```

```
C  MOVED OR REMOVED.
C
     IF (COMAND(:3) .EQ. 'MOV' .OR. COMAND(:3) .EQ. 'REM' ) THEN
       CALL JCOLOR(8)
       CALL JCMARK(5)
       CALL JMARK(X,Y)
       ENDIF
     IF (IFLG .EQ. 1) THEN
       DO 10  J = 1, NDEPL
       IF( ABS(XTAR(J,1)-X) .GT. XDX .OR. ABS(XTAR(J,2)-Y) .GT. YDY )
    +  GO TO 1103
       NMV = NMV+1
C
C  CHECK TO MAKE SURE IT IS A HUNDRED OR LESS ASSETS.
C
       IF( NMV .LE. 100 )  GO TO 1102
       PRINT*,'ONLY 100 AT A TIME'
       NMV = 100
       GO TO 1104
 1102   ITMP(NMV) = J
   10   CONTINUE
 1104   CONTINUE
       DO 20  KK = 1, NMV
       J = ITMP(KK)
       WRITE(NUMNUT,9) (XTAR(J,K),K=1,2)
    9  FORMAT(2F9.1)
C
C  THE FOLLOWING CHECKS FOR BLANKS IN KILL CRITERIA AND
C  POSTURE.
C
       IF( INDEX(NAME(J),' ') .EQ. 0 )  THEN
       NN = 19
       ITEMP = NAME(J)
       GO TO 1108
       ELSE
       ITEMP = ' '
       ID = NAME(J)
       NN = 0
       ENDIF
 1105   CONTINUE
       N1 = INDEX(ID,' ')
       IF( NN .EQ. 0 )  THEN
       ITEMP = ID(:N1)
       ELSE
       ITEMP = ITEMP(:NN) // ID(:N1)
       ENDIF
       NN = NN+N1
       ID = ID(N1+1:)
       IF( ID .NE. ' ' )  GO TO 1105
 1108   CONTINUE
       IDSYM = ' [ ],'
       DO 30  JK = 1, NVIS
       IF( ISYM(J) .EQ. IVIS(JK) )  GO TO 1118
```

```
  30   CONTINUE
C
C  IF HERE, ITS SYMBOL IS VISIBLE.
C
      IF( ISYM(J) .EQ. 'X' )  THEN
      IDSYM(3:3) = 'X'
      ELSE
      IDSYM(3:3) = ISYM(J)
      ENDIF
 1118   CONTINUE
      ILINE = ITEMP(:NN-1) // IDSYM // NUMNUT
      NINLN = NN + 24
      PRINT*,ILINE
      WRITE(ILINE,19) PAR(J,1),(IPAR(J,K),K=2,7)
  19   FORMAT( 3X,F5.2,6(1X,I2))
      PRINT*,ILINE
  20   CONTINUE
    ELSE
C
C  THE FOLLOWING PRINTS OUT WEAPON INFORMATION.
C
      NW = 0
      DO 40  J = 1, KRDS(NF)
        IF( ABS(XKA(J,1)-X) .GT. XDX .OR. ABS(XKA(J,2)-Y)
   +      .GT. YDY) GO TO 2110
        NMV = NMV+1
        WRITE(ILINE,29) KREP(J),KWPN(J),TIMK(J)
  29     FORMAT('REP=',I3,'. WPN#=',I3,'. TIM=',F5.0)
        PRINT*,ILINE
        DO 50 II = 1, NWTOTL
          IF (NOWPN(II) .EQ. KWPN(J).AND.NFIL(II) .EQ. KF(J))NW=II
  50     CONTINUE
        WRITE(ILINE,39) WPNAME(NW)
  39     FORMAT('NAME = ',A18)
        PRINT*,ILINE
        WRITE(ILINE,49) KWIN(J)
  49     FORMAT('WIND = ',F10.2)
        PRINT*,ILINE
        WRITE(ILINE,59) (XKA(J,L),L=1,3)
  59     FORMAT('AGZ=',3F8.1)
        PRINT*,ILINE
        WRITE(ILINE,69) (XKD(J,L),L=1,3)
  69     FORMAT('DGZ=',3F8.1)
        PRINT*,ILINE
  40   CONTINUE
    ENDIF
    IF( NMV .GT. 0 )  GO TO 1115
     PRINT*,'NOTHING THERE'
1115 PRINT*,'press return to continue'
    READ(*,79)I
79   FORMAT(A1)
    CALL JCLOSE
    RETURN

    END
```

SUBROUTINE LIST

FUNCTION: Subroutine LIST lists all weapon numbers and weapon names.

PARAMETERS: NONE

DISCUSSION: Subroutine LIST lists all weapon numbers and weapon names. This subroutine prints out the following information for all weapons: tape number for weapon file, the weapon number and the weapon name.

LIST VARIABLE GLOSSARY:

Variable	Definition	Type	Units
NFIL	Tape number to which weapon file is written.	Integer	N/A
NOWPN	Number of weapons under that name.	Integer	N/A
NWTOTL	Total number of weapons.	Integer	N/A
OLDFIL	Flags if weapon file has been read before.	Logical	N/A
WPNAME	Name of weapon from weapon deployment file.	Char.	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: NONE

Subroutines calling LIST: WPNMOD

```
C*******************************LIST*****************************************
      SUBROUTINE LIST
C
C     LIST IS CALLED BY WEAPON AND LISTS ALL WEAPON NUMBERS
C     AND WEAPON NAMES.
C
      COMMON/WPNDAT/KREP(NRDMAX),KWPN(NRDMAX),TIMK(NRDMAX),KF(NRDMAX),
     + XKA(NRDMAX,3),XKD(NRDMAX,3),KWIN(NRDMAX),KRDS(NFILES),NFIL(NWPNS)
     + ,NWEAPS(NFILES), IREP(NWPNS), NOWPN(NWPNS), WXMIN(NWPNS), NWPN,
     + WXMAX(NWPNS), WYMIN(NWPNS), WYMAX(NWPNS), WIND(NRDMAX),NWTOTL
      COMMON/WPNFILE/ WFILE(NFILES), NCLR(NFILES), NF
      COMMON/WPNCHR/ IWPN(NWPNS), WPNAME(NWPNS)
      CHARACTER*18  WPNAME, IWPN, LINE*50
      PARAMETER ( NRDMAX = 1150 )
      PARAMETER ( NWPNS = 20 )
      PARAMETER ( NFILES = 10 )
      LOGICAL OLDFIL
      REAL KWIN
      OLDFIL = .FALSE.
C
C  THIS SECTION PRINTS THE FOLLOWING INFORMATION FOR ALL
C  WEAPONS: TAPE # FOR WEAPON FILE, THE WEAPON NUMBER AND THE
C  WEAPON NAME.
C
      PRINT*,'TAPE #        WEAPON #      WEAPON NAME'
      PRINT*,'------        --------      -----------'
      PRINT*,' '
C
C  OLDFIL IS A LOGICAL TO SEE IF WEAPON FILE HAS BEEN READ IN.
C
      IF ( .NOT. OLDFIL ) THEN
        DO 10 I = 1, NWTOTL
          WRITE(LINE,9) NFIL(I), NOWPN(I), WPNAME(I)
          PRINT*,LINE
   10   CONTINUE
      ENDIF
      OLDFIL = .FALSE.
      PRINT*,' '
    9 FORMAT(3X,I3,10X,I3,13X,A18)
      RETURN
      END
```

SUBROUTINE LOCAT

FUNCTION: Subroutine LOCAT locates a particular X,Y coordinate on the screen.

PARAMETERS: NONE

DISCUSSION: Subroutine LOCAT locates a particular X,Y coordinate on the screen. The user inputs the x,y coordinate then the program places a marker on the screen corresponding to that location.

LOCAT VARIABLE GLOSSARY:

Variable	Definition	Type	Units
ILINE	Holds inputed X & Y value.	Char.	N/A
ITEMP	Temporarily holds X value or Y value.	Char.	N/A
N1	Holds value of the number of commas.	Integer	N/A
TXM	Minimum X value.	Real	Meters
TXP	Maximum X value.	Real	Meters
TYM	Minimum Y value.	Real	Meters
TYP	Maximum Y value.	Real	Meters

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: INDEX

Subroutines calling LOCAT: TARMOD WPNMOD

```
C**********************************LOCAT**************************************
      SUBROUTINE LOCAT
C
C  THE LOCATE OPTION LOCATES AN X,Y PAIR ON THE SCREEN.
C
      COMMON/PAD/TXM,TXP,TYM,TYP, LEFT, FACTOR, TOP,FXM,FXP,FYM,FYP
      CHARACTER*18 ITEMP,ILINE*40

 1800 ILINE = ' '
C
C  TXM AND TYM ARE MINIMUM COORDINATES OF THE SCREEN.
C
      PRINT*,TXM,TYM
C
C  PROMPT USER FOR X, Y COORDINATES TO LOCATE.
C
      PRINT*,'INPUT X,Y >'
      READ(*,9)ILINE
    9 FORMAT(A40)
C
C  CONVERTS STRING INTO X AND Y VALUES.
C
      N1 = INDEX(ILINE,',')
      IF( N1 .LE. 0 )  GO TO 1800
      ITEMP = ILINE(:N1-1)
      READ(ITEMP,19,ERR=7734) X
      ILINE=ILINE(N1+1:)
      READ(ILINE,19,ERR=7734) Y
   19 FORMAT( F10.3 )
C
C  CHECK TO SEE IF COORDINATES ARE WITHIN SCREEN LIMITS.
C
      IF( X .GT. TXP .OR. X .LT. TXM )  GO TO 1850
      IF( Y .GT. TYP .OR. Y .LT. TYM )  GO TO 1850
C
C  PLACES X AT THE X,Y COORDINATE IF IT IS ON THE SCREEN.
C
C  PVI ROUTINES TO OPEN A GPAPHICS WINDOW AND PLACE A MARKER AT
C  THE DESIGNATED POINT.
C
      CALL JOPEN
      CALL JCOLOR(2)
      CALL JCMARK(2)
      CALL JMARK(X,Y)
      CALL JCLOSE
      RETURN
 1850 CONTINUE
      PRINT*,'X,Y NOT ON SCREEN'
      PRINT*,'RETURN TO COMMAND MODE!'
      GO TO 9999
 7734 CONTINUE
 9999 RETURN
      END
```

SUBROUTINE MOVSUB

FUNCTION: Subroutine MOVSUB moves or removes assets from the deployment.

PARAMETERS: IOPT flags which option to implement. IRECALC flags to recalculate the initial screen coordinates.

DISCUSSION: Subroutine MOVSUB moves or removes assets from the deployment. This subroutine calls IDENTFY to locate the assets at a given location. MOVSUB also calls REFRESH to redraw the screen.

MOVSUB VARIABLE GLOSSARY:

Variable	Definition	Type	Units
ALTPOS	Holds alternate posture values.	Char.	N/A
COMM	Holds commented deployment line.	Char.	N/A
ID	Holds asset name to change.	Char.	N/A
IPAR	Array of kill criteria & initial postures.	Integer	N/A
ISYM	Holds character for symbol.	Char.	N/A
IT	Set to value of ITMP.	Integer	N/A
ITMP	Number of deployed asset points.	Integer	N/A
NAME	Asset names from input file.	Char.	N/A
NDEPL	Number of deployment points.	Integer	N/A
NIN	Counter to rearrange assets.	Integer	N/A
NMV	Number of assets at a deployment point.	Integer	N/A
NOUT	Counter for ITMP values.	Integer	N/A
XTAR	Array of X and Y coordinates of asset deployment.	Real	Meters

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: IDENTIFY REFRESH

System Library functions: NONE

Subroutines calling MOVSUB: TARMOD

```
C*******************************MOVSUB*****************************************
      SUBROUTINE MOVSUB(IOPT,IRECALC)
C
C  THIS ROUTINE MOVES OR REMOVES ASSETS FROM THE DEPLOYMENT.
C
      COMMON/TEMPS/ ITMP
      COMMON/CHRTMP/ ID
      COMMON/NAMES/NAME(NDEPMX),ISYM(NDEPMX),ICOLOR(NDEPMX),
     + ALTPOS(NDEPMX), COMM(NDEPMX)
      COMMON/DEPLOY/ NDEPL, XTAR(NDEPMX,2), IPAR(NDEPMX,7)
      CHARACTER*18 NAME,ID, ALTPOS*40, COMM*40
      CHARACTER*1 ISYM, ICH, ICOLOR
      PARAMETER ( NDEPMX = 702 )
      DIMENSION ITMP(100)
      DIMENSION PAR(NDEPMX,7)
      EQUIVALENCE ( PAR,IPAR )

 2102 CONTINUE
      PRINT*,'PUT CURSOR ON ITEM'
C
C  CALLS IDENTFY TO SEE WHAT ASSETS ARE AT THAT LOCATION.
C
      CALL IDENTFY(NMV,1)
C
C  IF MORE THAN ONE ITEM, FIND OUT WHICH TO MOVE.
C
      IF( NMV .EQ. 1 )  GO TO 2120
 2112 CONTINUE
      PRINT*,'WHICH ONE OR ALL? >'
      READ(*,9)ID
    9 FORMAT(A18)
C
C  MOVE ALL ASSETS AT THAT POINT.
C
      IF( ID .EQ. 'ALL' )  GO TO 2120
       DO 10  KK = 1, NMV
       IT=ITMP(KK)
C
C  MOVES ONLY ONE ASSET FROM THAT POINT.
C
      IF( NAME(IT) .EQ. ID )  GO TO 2116
   10 CONTINUE
      GO TO 2112
 2116 NMV = 1
      ITMP(1) = IT
 2120 CONTINUE
C
C  CONFIRMS THE RIGHT ASSETS TO MOVE.
C
      PRINT*,'O.K.? Y OR N? >'
      READ(*,19)ICH
   19 FORMAT(A1)
C
```

```
C  REFRESH IS CALLED TO REDRAW SCREEN.
C
     IF( ICH .NE. 'Y' )  CALL REFRESH
     IF ( IOPT .EQ. 2 ) THEN
C
C  RECALL THAT ITMPS (ASSET NUMBER) ARE IN ASCENDING ORDER. THIS
C  FINDS THE RIGHT ASSETS TO REMOVE.
C
 2124    NIN = 0
         NOUT = 1
         DO 20  J = 1, NDEPL
C
C  FLAG TO GET OUT AFTER ALL THE ASSETS THAT NEED TO BE REMOVED
C  HAVE BEEN REMOVED.
C
         IF( J .EQ. ITMP(NOUT) )  GO TO 2128
         NIN = NIN+1
         NAME(NIN)=NAME(J)
         ALTPOS(NIN) = ALTPOS(J)
         COMM(NIN) = ALTPOS(J)
         ISYM(NIN)=ISYM(J)
         XTAR(NIN,1)=XTAR(J,1)
         XTAR(NIN,2)=XTAR(J,2)
         DO 30 K=1, 7
   30    IPAR(NIN,K)=IPAR(J,K)
         GO TO 2130
 2128    IF( NOUT .LT. NMV )  NOUT = NOUT+1
   20    CONTINUE
C
C  NOW CLEAR STORAGE RECENTLY VACATED - IF NOT RESET BY FUTURE INPUT.
C
         NDEPL = NIN
         IRECALC = 1
      ELSE
C
C  MOVE. MERELY CHANGE THE XTARS TO X,Y FROM CURSOR
C  PVI ROUTINE PLACES A MARKER AT THE NEW LOCATION OF THE MOVED
C  ASSETS.
C
         CALL JLOCAT(1,1,1,B,XZ,YZ)
         CALL JCONVW(XZ,YZ,X,Y,Z)
         CALL JOPEN
         CALL JCOLOR(1)
         CALL JCMARK(5)
         CALL JMARK(X,Y)
         CALL JCLOSE
         DO 40  J = 1, NMV
         IT = ITMP(J)
         XTAR(IT,1) = X
         XTAR(IT,2) = Y
 40      CONTINUE
         IRECALC = 1
      ENDIF
```

```
RETURN
END
```

PROGRAM MAIN

FUNCTION: Program MAIN is the main routine for the AURATEK program. It links the main subroutines together.

PARAMETERS: NONE

DISCUSSION: Program MAIN is used to link subroutines AURATEK, TARMOD and ATEKOUT together.

HEADER VARIABLE GLOSSARY:

Variable	Definition	Type	Units
---	---	---	---

NO SUBROUTINE VARIABLES.

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: AURATEK, TARMOD, ATEKOUT

System Library functions: NONE

Subroutines calling HEADER: NONE

```
C****************************** MAIN *********************************

      PROGRAM MAIN
      CALL AURATEK
      CALL TARMOD
      CALL ATEKOUT
      STOP
      END
```

SUBROUTINE PATTERN

FUNCTION: Subroutine PATTERN calculates the points needed to plot the elliptical pattern to illustrate chemical and conventional weapon effects.

PARAMETERS: CALL PATTERN(I, XKA, XMAX, XMIN, YMAX, YMIN, WIND)

DISCUSSION: Subroutine PATTERN calculates the points needed to plot the elliptical pattern to illustrate chemical weapon effects. PATTERN works by using the axis length and an incremental angle to find the points needed to plot the ellipse.

PATTERN VARIABLE GLOSSARY:

Variable	Definition	Type	Units
A	Maximum value of XSEMI and YSEMI.	Real	N/A
B	Minimum value of XSEMI and YSEMI.	Real	N/A
C1	Cosine of the value WIND1.		Real
C2	Cosine of the value P.	Real	N/A
C3	Value of T1.	Real	Radians
H	X coordinate offset for weapon pattern.	Real	Meters
K	Y coordinate offset for weapon pattern.	Real	Meters
P	Increment angle for ellipse.	Real	Radians
S1	Sine of the value WIND1.	Real	N/A
S2	Sine of the value P.	Real	N/A
S3	New increment value for sine angle.	Real	Radians
T1	Calculates new increment for cosine angle.	Real	Radians
WIND1	Value of wind converted to radians.	Real	N/A
X1	Long axis of ellipse.	Real	Meters
XKA	Array of X & Y coordinates to mark weapon.	Real	N/A
XPAT	X point used to plot pattern.	Real	N/A
XSEMI	Half of length in X direction.	Real	N/A
Y1	Short angle of ellipse.	Real	Meters
YPAT	Y point used to plot pattern.	Real	N/A
YSEMI	Half of length in Y direction.	Real	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: AMAX1, AMIN1, SIN, COS

Subroutines calling PATTERN: EMPLOY

```
C*************************PATTERN*****************************************
      SUBROUTINE PATTERN(I,XKA,XMAX,XMIN,YMAX,YMIN,WIND)
C
C  PATTERN IS CALLED BY EMPLOY.  IT CALCULATES THE POINTS NEEDED
C  TO PLOT THE ELLIPTICAL PATTERN WHICH IS USED TO ILLUSTRATE THE
C  CHEMICAL AND CONVENTIONAL WEAPON EFFECTS.
C
      COMMON/POINTS/ XPAT(NPTS), YPAT(NPTS)
      PARAMETER (NRDMAX = 1150 )
      PARAMETER (NPTS = 30)
      DIMENSION XKA(NRDMAX,3)
      REAL K
C
C  CHANGES WIND ANGLE FROM DEGREES TO RADIANS.
C
      WIND1 = WIND/57.2957795
C
C  FINDS HALF-WAY DISTANCES TO FIGURE ELLIPSE DIMENSION.
C
      XSEMI = (XMAX - XMIN)/2.0
      YSEMI = (YMAX - YMIN)/2.0
C
C  X AND Y COORDINATES FOR START OF ELLIPSE PLOT.
C
      H = XKA(I,1) + (XSEMI + XMIN) * COS(WIND1)
      K = XKA(I,2) + (XSEMI + XMIN) * SIN(WIND1)
C
C  A IS THE LONG AXIS VALUE FOR THE ELLIPSE.
C  B IS THE SHORT AXIS VALUE FOR THE ELLIPSE.
C
      A = AMAX1(XSEMI,YSEMI)
      B = AMIN1(XSEMI,YSEMI)
C
C  ANGLE TO INCREMENT EACH TIME FOR X AND Y POINTS OF ELLIPSE.
C
      P = 2.0 * 3.14156/ (NPTS-1)
C
C  COSINE AND SINE VALUES OF WIND1.
C
      C1 = COS(WIND1)
      S1 = SIN(WIND1)
C
C  COSINE AND SINE VALUES OF P.
C
      C2 = COS(P)
      S2 = SIN(P)
C
C  INITIAL VALUES.
C
      C3 = 1.0
      S3 = 0.0
C
C  DO LOOP TO PLOT POINTS.
```

```
C
      DO 10  M = 1,NPTS
        X1 = A*C3
        Y1 = B*S3
        XPAT(M) = H + X1*C1 - Y1*S1
        YPAT(M) = K + X1*S1 + Y1*C1
        T1 = C3*C2 - S3*S2
        S3 = S3*C2 + C3*S2
        C3 = T1
   10 CONTINUE
      RETURN
      END
```

SUBROUTINE PREVIUS

FUNCTION: Subroutine PREVIUS restores the screen to the previous view.

PARAMETERS: NONE

DISCUSSION: Subroutine PREVIUS restores the screen to the previous view. REFRESH is called to redraw the screen using the previous values.

PREVIUS VARIABLE GLOSSARY:

Variable	Definition	Type	Units
REDRAW	Flag to see if screen needs to be redrawn.	Logical	N/A
TXM	Minimum value of XZ(1) and XZ(2).	Real	Meters
TXP	Maximum value of XZ(1) and XZ(2).	Real	Meters
TYM	Minimum value of YZ(1) and YZ(2).	Real	Meters
TYP	Maximum value of YZ(1) and YZ(2).	Real	Meters
XZ	X world coordinate used in PVI calls.	Real	Meters
XXM	Holds value TXM.	Real	Meters
XXP	Holds value TXP.	Real	Meters
XYM	Holds value TYM.	Real	Meters
XYP	Holds value TYP.	Real	Meters

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: REFRESH

System Library functions: NONE

Subroutines calling PREVIUS: TARMOD

```
C*****************************PREVIUS*****************************************
      SUBROUTINE PREVIUS
C
C  PREVIOUS RESTORES THE SCREEN TO THE PREVIOUS VIEW.
C
      COMMON/SCRDAT/ XXM, XXP, XYM, XYP, TXM1, TXP1, TYM1, TYP1
      COMMON/FLAG/ REDRAW
      COMMON/PAD/TXM,TXP,TYM,TYP, LEFT, FACTOR, TOP,FXM,FXP,FYM,FYP
      LOGICAL REDRAW
C
C  SET CURRENT SCREEN VALUES TO THE PREVIOUS VALUES.
C
      TXM = XXM
      TXP = XXP
      TYM = XYM
      TYP = XYP
C
C  REDRAWS THE SCREEN.
C
      CALL REFRESH
      REDRAW = .TRUE.
      RETURN
      END
```

SUBROUTINE REFRESH

FUNCTION: Subroutine REFRESH redraws the screen.

PARAMETERS: NONE

DISCUSSION: Subroutine REFRESH redraws the screen. REFRESH takes each deployment point and plots a marker with the appropriate symbol and color.

REFRESH VARIABLE GLOSSARY:

Variable	Definition	Type	Units
ASSETS	Flag used in invisible option.	Logical	N/A
BORDER	Turns border on or off around the work area.	Logical	N/A
DUMMY	Flag used in invisible option.	Logical	N/A
DX	One percent of X length.	Real	Meters
DY	One percent of Y length.	Real	Meters
ICOLOR	Sets color for symbol read from the input file.	Char.	N/A
NAME	Asset name.	Char.	N/A
NDEPL	Number of deployment points.	Integer	N/A
NEWSCR	Flags if a new screen needs to be drawn.	Logical	N/A
REDRAW	Flag to see if screen needs to be redrawn.	Logical	N/A
TXM	Minimum value of XZ(1) and XZ(2).	Real	Meters
TXM1	Minimum X value used in PVI.	Real	Meters
TXP	Maximum value of XZ(1) and XZ(2).	Real	Meters
TXP1	Maximum X value used in PVI.	Real	Meters
TYM	Minimum value of YZ(1) and YZ(2).	Real	Meters
TYM1	Minimum Y value used in PVI.	Real	Meters
TYP	Maximum value of YZ(1) and YZ(2).	Real	Meters
TYP1	Maximum Y value used in PVI.	Real	Meters
XTAR	Array of x and y coordinates of asset deployment.	Real	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: NONE

Subroutines calling REFRESH: EMPLOY FULLSC INVIS MOVSUB PREVIUS
RESCALE SETSCR SYMCOL TARMOD WPNMOD ZOOM

```
C*****************************REFRESH*****************************************
      SUBROUTINE REFRESH
C
C     REFRESH REDRAWS THE SCREEN.
C
      COMMON/SCRDAT/ XXM, XXP, XYM, XYP, TXM1, TXP1, TYM1, TYP1
      COMMON/TMAX/ TYMAX, TYMIN
      COMMON/FLAG/ REDRAW
      COMMON/BRDR/ BORDER
      COMMON/LOGS/ NEWSCR, DEFAULT, DUMMY, ASSETS, INITDEP
      COMMON/NAMES/NAME(NDEPMX),ISYM(NDEPMX),ICOLOR(NDEPMX),
     + ALTPOS(NDEPMX), COMM(NDEPMX)
      COMMON/PAD/TXM,TXP,TYM,TYP, LEFT, FACTOR, TOP,FXM,FXP,FYM,FYP
      COMMON/VISBL/ IVIS(64)
      COMMON/VISINT/ NVIS
      COMMON/DEPLOY/ NDEPL, XTAR(NDEPMX,2), IPAR(NDEPMX,7)
      CHARACTER*1 ICOLOR, ISYM, IVIS, NAME*18, ALTPOS*40, COMM*40
      PARAMETER ( NDEPMX = 702 )
      LOGICAL NEWSCR,  DUMMY, ASSETS, REDRAW, BORDER
      DIMENSION PAR(NDEPMX,7)
      EQUIVALENCE ( PAR,IPAR )
      REAL LEFT
C
C     FIRST, RESET VIRTUAL WINDOW.
C

      NEWSCR = .TRUE.
      DX = 0.01*(TXP-TXM)
      DY = 0.01*(TYP-TYM)
      TXM1 = TXM-DX
      TXP1 = TXP+DX
      TYM1 = TYM-DY
      TYP1 = TYP+DY

C
C  PVI ROUTINE TO DEFINE THE BOUNDARIES OF THE VIEWPLANE.
C
      CALL JWINDO(TXM1,TXP1,TYM1,TYP1)

C
C  DRAW BORDER AROUND DEPLOYMENT AREA.  THIS PREVENTS
C  USER FROM ACCIDENTLY GOING OUT OF THE WORK AREA WITH
C  CURSORS.
C
C  PVI ROUTINES OPEN A GRAPHICS WINDOW AND FRAMES EVERYTHING IN IT.
C
      CALL JFRAME
      CALL JOPEN
      IF (BORDER) THEN
C
C  PVI ROUTINES TO SET THE COLOR AND DRAW THE BORDER AROUND THE
C  SCREEN.
C
```

```
        CALL JCOLOR(2)
        CALL JMOVE(TXM1,TYM1)
        CALL JDRAW(TXM1,TYP1)
        CALL JDRAW(TXP1,TYP1)
        CALL JDRAW(TXP1,TYM1)
        CALL JDRAW(TXM1,TYM1)
        CALL JCOLOR(6)
      ENDIF
C
C  LOOP 10 GOES THROUGH EACH DEPLOYMENT AND PLOTS A MARKER.
C  IMPLEMENTS THE INVISIBLE OPTION.
C
      DO 10  J = 1, NDEPL

      IF( XTAR(J,1) .GT. TXP .OR. XTAR(J,1) .LT. TXM )  GO TO 2090
      IF( XTAR(J,2) .GT. TYP .OR. XTAR(J,2) .LT. TYM )  GO TO 2090
      IF ((.NOT. (DUMMY)) .AND. (PAR(J,1) .LT. 0)) GO TO 2090
      IF ((.NOT. (ASSETS)) .AND. (PAR(J,1) .GT. 0)) GO TO 2090
      DO 20  K = 1, NVIS
      IF( ISYM(J) .EQ. IVIS(K) )  GO TO 2090
   20 CONTINUE
C
C  THE NEXT BLOCK OF CODE, CHOOSES THE APPROPRIATE COLOR FOR
C  THE CHOSEN MARKER.
C
      CALL JCOLOR(7)
      IF (ICOLOR(J) .EQ. 'W') THEN
        CALL JCOLOR(7)
      ELSE IF (ICOLOR(J) .EQ. '1') THEN
        CALL JCOLOR(1)
      ELSE IF (ICOLOR(J) .EQ. '2') THEN
        CALL JCOLOR(2)
      ELSE IF (ICOLOR(J) .EQ. '3') THEN
        CALL JCOLOR(3)
      ELSE IF (ICOLOR(J) .EQ. '4') THEN
        CALL JCOLOR(4)
      ELSE IF (ICOLOR(J) .EQ. '5') THEN
        CALL JCOLOR(5)
      ELSE IF (ICOLOR(J) .EQ. '6') THEN
        CALL JCOLOR(6)
      ELSE IF (ICOLOR(J) .EQ. '7') THEN
        CALL JCOLOR(7)
      ENDIF
      CALL JCMARK(5)
      IF (ISYM(J) .EQ. '1') THEN
        CALL JCMARK(1)
      ELSE IF (ISYM(J) .EQ. '2') THEN
        CALL JCMARK(2)
      ELSE IF (ISYM(J) .EQ. '3') THEN
        CALL JCMARK(3)
      ELSE IF (ISYM(J) .EQ. '4') THEN
        CALL JCMARK(4)
      ELSE IF (ISYM(J) .EQ. '5') THEN
```

```
      CALL JCMARK(5)
    ENDIF
    CALL JMARK( XTAR(J,1) ,XTAR(J,2) )
10  CONTINUE
    REDRAW = .TRUE.
    CALL JCLOSE
    RETURN
    END
```

SUBROUTINE SCREEN

FUNCTION: Subroutine SCREEN displays the current screen coordinates.

PARAMETERS: NONE

DISCUSSION: Subroutine SCREEN displays the current screen coordinates.

SCREEN VARIABLE GLOSSARY:

Variable	Definition	Type	Units
ILINE	String containing screen coordinates.	Char.	N/A
TXM	Minimum value of XZ(1) and XZ(2) from ZOOM.	Real	Meters
TXP	Maximum value of XZ(1) and XZ(2) from ZOOM.	Real	Meters
TYM	Minimum value of YZ(1) and YZ(2) from ZOOM.	Real	Meters
TYP	Maximum value of YZ(1) and YZ(2) from ZOOM.	Real	Meters

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: NONE

Subroutines calling SCREEN: SETSCR TARMOD WPNMOD

```
C*****************************SCREEN*************************************
      SUBROUTINE SCREEN
C
C  SCREEN DISPLAYS THE CURRENT SCREEN COORDINATES.
C
      COMMON/PAD/ TXM,TXP,TYM,TYP,LEFT,FACTOR,TOP,FXM,FXP,FYM,FYP
      CHARACTER*40 ILINE
      REAL LEFT
C
C  PRINTS THE CURRENT SCREEN DIMENSIONS.
C
      PRINT*,'XMIN,XMAX,YMIN,YMAX'
      WRITE(ILINE,9) TXM,TXP,TYM,TYP
    9 FORMAT(4F10.1)
      PRINT*,ILINE
      RETURN
      END
```

SUBROUTINE SEARCH

FUNCTION: Subroutine SEARCH looks for all assets in the deployment containing a specified string.

PARAMETERS: NONE

DISCUSSION: Subroutine SEARCH looks for assets containing a specified string. SEARCH works by taking a string of an asset searching through the deployment for a match. SEARCH will find all occurrences of that string.

SEARCH VARIABLE GLOSSARY:

Variable	Definition	Type	Units
ID	Holds string to find.	Char.	N/A
IEND	Holds number of blanks found in ID.	Integer	N/A
NAME	Asset names from input file.	Char.	N/A
NDEPL	Number of deployment points.	Integer	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: INDEX

Subroutines calling SEARCH: TARMOD

```
C************************SEARCH*****************************************
      SUBROUTINE SEARCH
C
C  SEARCH LOOKS FOR ASSETS CONTAINING A SPECIFIED STRING.
C
      COMMON/FLAG/ REDRAW
      COMMON/NAMES/NAME(NDEPMX),ISYM(NDEPMX),ICOLOR(NDEPMX),
     + ALTPOS(NDEPMX), COMM(NDEPMX)
      COMMON/DEPLOY/ NDEPL, XTAR(NDEPMX,2), IPAR(NDEPMX,7)
      CHARACTER*18 NAME,ID, ALTPOS*40, COMM*40
      CHARACTER*1 ISYM, ICOLOR
      PARAMETER ( NDEPMX = 702 )
      DIMENSION PAR(NDEPMX,7)
      LOGICAL REDRAW
      EQUIVALENCE ( PAR,IPAR )
C
C  ASK FOR STRING TO SEARCH FOR.
C
      PRINT*,'ENTER STRING >'
      READ(*,9)ID
    9 FORMAT(A18)
      IEND = INDEX(ID,' ')
      IEND = IEND - 1
      DO 10 I = 1, NDEPL
        DO 20 J = 1, 18-IEND+1
          IF (NAME(I)(J:J+IEND-1) .EQ. ID(:IEND)) THEN
            PRINT*,NAME(I)
          ENDIF
   20   CONTINUE
   10 CONTINUE
      REDRAW = .FALSE.
      RETURN
      END
```

SUBROUTINE SETSCR

FUNCTION: Subroutine SETSCR allows the user to set the screen coordinates.

PARAMETERS: NONE

DISCUSSION: Subroutine SETSCR allows the user to set the screen coordinates. First, SCREEN is called to show the user the dimensions of the current screen. Next, REFRESH is called to redraw the screen at the new coordinates.

SETSCR VARIABLE GLOSSARY:

Variable	Definition	Type	Units
EX	Difference between X maximum and X minimum values.	Real	Meters
EY	Difference between Y maximum and Y minimum values.	Real	Meters
ILINE	String that holds maximum and minimum coordinates.	Char.	N/A
ITEMP	Temporarily holds each coordinate.	Char.	N/A
N1	Holds number of commas counted in ILINE.	Integer	N/A
REDRAW	Flags if screen needs to be redrawn.	Logical	N/A
TXM	Minimum. value of XZ(1) and XZ(2) from ZOOM.	Real	Meters
TXP	Maximum value of XZ(1) and XZ(2) from ZOOM.	Real	Meters
TYM	Minimum value of YZ(1) and YZ(2) from ZOOM.	Real	Meters
TYP	Maximum value of YZ(1) and YZ(2) from ZOOM.	Real	Meters
XXM	Holds value TXM.	Real	Meters
XXP	Holds value TXP.	Real	Meters
XYM	Holds value TYM.	Real	Meters
XYP	Holds value TYP.	Real	Meters

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: REFRESH SCREEN

System Library functions: INDEX

Subroutines calling SETSCR: TARMOD

```
C*****************************SETSCR*****************************************
      SUBROUTINE SETSCR
C
C  SET SCREEN ALLOWS THE USER TO SET THE SCREEN COORDINATES.
C
      COMMON/WIDTH/ EX, EY
      COMMON/FLAG/ REDRAW
      COMMON/PAD/TXM,TXP,TYM,TYP, LEFT, FACTOR, TOP,FXM,FXP,FYP
      CHARACTER*18 ITEMP,ILINE*40
      LOGICAL REDRAW


      REAL LEFT
      XXM = TXM
      XXP = TXP
      XYM = TYM
      XYP = TYP
C
C  SHOWS CURRENT SCREEN CORNERS.
C
      PRINT*,'current screen is'
      CALL SCREEN
 1700 ILINE = ' '
C
C ENTER NEW COORDINATES.
C
      PRINT*,'XMIN,XMAX,YMIN,YMAX >'
      READ(*,9)ILINE
    9 FORMAT(a40)
      N1 = INDEX(ILINE,',')
      IF( N1 .LE. 0 )  GO TO 1700
      ITEMP = ILINE(:N1-1)
      READ(ITEMP,19,ERR=7734) TXM
      ILINE = ILINE(N1+1:)
   19 FORMAT( F10.3 )
      N1 = INDEX(ILINE,',')
      IF( N1 .LE. 0 )  GO TO 1700
      ITEMP = ILINE(:N1-1)
      READ(ITEMP,1709,ERR=7734) TXP
      ILINE = ILINE(N1+1:)
      N1 = INDEX(ILINE,',')
      IF( N1 .LE. 0 )  GO TO 1700
      ITEMP = ILINE(:N1-1)
      READ(ITEMP,1709,ERR=7734) TYM
      ILINE = ILINE(N1+1:)
      READ(ILINE,1709,ERR=7734) TYP
      EX = TXP-TXM
      EY = TYP-TYM
C
C  WE MUST NOW CONTINUE INTO REFRESH TO SET VIRTUAL WINDOW.
C
      IF( EX .GT. 1.0E-6 .AND. EY .GT. 1.0E-6 )  THEN
        CALL REFRESH
```

```
      REDRAW = .TRUE.
    ELSE
     PRINT*,'RANGES MUST BE .GT. 0.'
      REDRAW = .FALSE.
    ENDIF
    GO TO 9999
7734 CONTINUE
    PRINT*,'INPUT ERROR!  TRY AGAIN!'
9999 RETURN
    END
```

SUBROUTINE SYMCOL

FUNCTION: Subroutine SYMCOL allows the user to designate a symbol or color for a particular target of group of targets.

PARAMETERS: NONE

DISCUSSION: Subroutine SYMCOL allows the user to designate a symbol or color for a particular target group of targets. REFRESH is called to redraw the screen with the new symbols and colors.

SYMCOL VARIABLE GLOSSARY:

Variable	Definition	Type	Units
IANS	Input of what type of string to change.	Char.	N/A
ICLR	Holds number of selected color.	Char.	N/A
ICOLOR	Holds ICLR.	Char.	N/A
ID	Holds asset name to change.	Char.	N/A
IEND	Counts number of blanks in ID.	Integer	N/A
ISY	Holds number of selected symbol.	Char.	N/A
ISYM	Holds ISY.	Char.	N/A
NAME	Asset names from input file.	Char.	N/A
NDEPL	Number of deployment points.	Integer	N/A
REDRAW	Flags if screen needs to be redrawn.	Logical	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: REFRESH

System Library functions: INDEX

Subroutines calling SYMCOL: TARMOD

```
C*****************************SYMCOL*****************************************
      SUBROUTINE SYMCOL
C
C  SYMCOL ALLOWS THE USER TO DESIGNATE A SYMBOL OR COLOR FOR A
C  PARTICULAR TARGET OR GROUP OF TARGETS.
C
      COMMON/FLAG/ REDRAW
      COMMON/NAMES/NAME(NDEPMX),ISYM(NDEPMX),ICOLOR(NDEPMX),
     + ALTPOS(NDEPMX), COMM(NDEPMX)
      COMMON/DEPLOY/ NDEPL, XTAR(NDEPMX,2), IPAR(NDEPMX,7)
      CHARACTER*18 NAME,ID, ALTPOS*40, COMM*40
      CHARACTER*1 ISYM, ISY, ICLR, ICOLOR, IANS
      PARAMETER ( NDEPMX = 702 )
      DIMENSION PAR(NDEPMX,7)
      LOGICAL REDRAW
      EQUIVALENCE ( PAR,IPAR )
      ISY = ' '
      ICLR = ' '
C
C  PRINT LIST OF SYMBOLS AND COLORS.
C  PROMPT FOR INPUT.
C
        PRINT*,'SYMBOL        COLOR'
        PRINT*,'--------------------'
        PRINT*,'1 -  .     1 - Red'
        PRINT*,'2 -  +     2 - Green'
        PRINT*,'3 -  *     3 - Yellow'
        PRINT*,'4 -  O     4 - Blue'
        PRINT*,'5 -  X     5 - Purple'
        PRINT*,'           6 - Cyan'
        PRINT*,'           7 - White'
        PRINT*,' '
      PRINT*,'ENTER SYMBOL >'
      READ(*,9)ISY
    9 FORMAT(a1)
      PRINT*,'ENTER COLOR >'
      READ(*,19)ICLR
   19 FORMAT(A1)
      PRINT*,' '
      PRINT*,'ENTER STRING/NAME > '
      READ(*,29)ID
   29 FORMAT(A18)
      IEND = INDEX(ID,' ')
      IEND = IEND -1
      PRINT*,'ENTER MODE FOR SYMBOL/COLOR CHANGE:'
      PRINT*,'  ALL NAMES (S)TARTING WITH STRING?'
      PRINT*,'  ALL NAMES (C)ONTAINING STRING?'
      PRINT*,'  ALL NAMES (I)DENTICAL TO STRING?'
      PRINT*,' '
      PRINT*,'ENTER "S" OR "C" OR "I" >'
      READ(*,39)IANS
   39 FORMAT(A1)
C
```

```
C  CHECK VARIABLE.
C  SET VARIABLES ACCORDINGLY FOR THAT STRING.
C
     IF (IANS .EQ. 'S') THEN
       DO 10 I = 1, NDEPL
         IF (NAME(I)(:IEND) .EQ. ID(:IEND)) THEN
           ISYM(I) = ISY
           ICOLOR(I) = ICLR
         ENDIF
  10   CONTINUE
     ELSE IF (IANS .EQ. 'C') THEN
       DO 20 I = 1, NDEPL
         DO 30 J = 1, 18-IEND+1
           IF (NAME(I)(J:J+IEND-1) .EQ. ID(:IEND)) THEN
             ISYM(I) = ISY
             ICOLOR(I) = ICLR
           ENDIF
  30     CONTINUE
  20   CONTINUE
     ELSE IF (IANS .EQ. 'I') THEN
       DO 40 I = 1, NDEPL
         IF (NAME(I) .EQ. ID) THEN
           ISYM(I) = ISY
           ICOLOR(I) = ICLR
         ENDIF
  40   CONTINUE
     ENDIF
     REDRAW = .FALSE.
     CALL REFRESH
     RETURN
     END
```

SUBROUTINE TARMOD

FUNCTION: Subroutine TARMOD provides the user with the main menu to access AURATEK options.

PARAMETERS: NONE

DISCUSSION: Subroutine TARMOD provides the user with the main menu to access AURATEK options. The user inputs the first three letters of the command and TARMOD calls the appropriate subroutine.

TARMOD VARIABLE GLOSSARY:

Variable	Definition	Type	Units
ASSETS	Flag used in INVIS subroutine.	Logical	N/A
COMAND	String that holds input command.	Char.	N/A
DEFAULT	Flags if default values are to be used.	Logical	N/A
DUMMY	Flag used in INVIS subroutine.	Logical	N/A
EX	Length in X direction.	Real	N/A
EY	Length in Y direction.	Real	N/A
FXM	Maximum value of X coordinate for screen.	Real	N/A
FXP	Minimum value of X coordinate for screen.	Real	N/A
FYM	Maximum value of Y coordinate for screen.	Real	N/A
FYP	Minimum value of Y coordinate for screen.	Real	N/A
INITDEP	Flags the first time a deployment has been used.	Logical	N/A
NCLR	Number of weapon files read in.	Integer	N/A
NDEPL	Number of deployment points.	Integer	N/A
NEWSCR	Flags if a new screen needs to be drawn.	Logical	N/A
NF	Number of weapon files used.	Integer	N/A
NFIL	Unit number of weapon file.	Integer	N/A
NWEAPS	Keeps count of weapons.	Integer	N/A
NWPNS	Number of weapons at that point.	Integer	N/A
NWTOTL	Total number of weapons read in.	Integer	N/A
REDRAW	Flags if screen needs to be redrawn.	Logical	N/A

TXM	Set to value of FXM.	Real	Meters
TXM1	Set to value of FXM.	Real	Meters
TXP	Set of value of FXP.	Real	Meters
TXP1	Set of value of FXP.	Real	Meters
TYM1	Set of value of FYM.	Real	Meters
TYP1	Set of value of FYM.	Real	Meters
USRPAT	Flags if USRPAT has been called before.	Logical	Meters
WFILE	Unit number for the file name.	Real	N/A
WPNAME	Weapon name from weapon input file.	Char.	N/A
WXM	Maximum screen coordinate in X direction.	Real	N/A
WXP	Minimum screen coordinate in X direction.	Real	N/A
WYM	Maximum screen coordinate in Y direction.	Real	N/A
WYP	Minimum screen coordinate in Y direction.	Real	N/A
XTAR	Array of X and Y coordinates of asset deployment.	Real	N/A
XXM	Set to the value of FXM.	Real	N/A
XXP	Set to the value of FXP.	Real	N/A
XYM	Set to the value of FYM.	Real	N/A
XYP	Set to the value of FYP.	Real	N/A

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: BULK COORDS DEPLOYA DISTANC ENDTEK FULLSC
FULLSC GRID HEADER HELP IDENTIFY INVIS LOCAT
MOVSUB PREVIUS REFRESH RESCALE SCREEN SEARCH
SETSCR SYMCOL WPNMOD ZOOM

System Library functions: AMIN1, AMAX1

Subroutines calling TARMOD: MAIN

```
C**************************TARMOD*********************************************
      SUBROUTINE TARMOD
C
C  TARGET MODE IS THE MAIN GRAPHICS ROUTINE.  IT QUERIES THE
C  USER AS TO THE OPTIONS DESIRED AND CALLS THE APPROPRIATE
C  ROUTINES TO IMPLEMENT THESE OPTIONS.
C
      COMMON/WPNFILE/ WFILE(NFILES), NCLR(NFILES), NF
      COMMON/WIDTH/ EX, EY
      COMMON/BRDR/ BORDER
      COMMON/COMAND/COMAND
      COMMON/PTN/ USRPAT(NWPNS)
      COMMON/FLAG/ REDRAW
      COMMON/SCRDAT/ XXM, XXP, XYM, XYP, TXM1, TXP1, TYM1, TYP1
      COMMON/LOGS/ NEWSCR, DEFAULT, DUMMY, ASSETS, INITDEP
      COMMON/WDIM/ WXM, WXP, WYM, WYP
      COMMON/PAD/TXM,TXP,TYM,TYP, LEFT, FACTOR, TOP,FXM,FXP,FYM,FYP
      COMMON/VISBL/ IVIS(64)
      COMMON/VISINT/ NVIS
      COMMON/DEPLOY/ NDEPL, XTAR(NDEPMX,2), IPAR(NDEPMX,7)
      COMMON/WPNDAT/KREP(NRDMAX),KWPN(NRDMAX),TIMK(NRDMAX),KF(NRDMAX),
     + XKA(NRDMAX,3),XKD(NRDMAX,3),KWIN(NRDMAX),KRDS(NFILES),NFIL(NWPNS)
     + ,NWEAPS(NFILES), IREP(NWPNS), NOWPN(NWPNS), WXMIN(NWPNS), NWPN,
     + WXMAX(NWPNS), WYMIN(NWPNS), WYMAX(NWPNS), WIND(NRDMAX),NWTOTL
      COMMON/WPNCHR/ IWPN(NWPNS), WPNAME(NWPNS)
      CHARACTER*18 COMAND,REPEAT,IVIS*1, ANS*1, WPNAME, IWPN
      PARAMETER ( NDEPMX = 702 )
      PARAMETER ( NWPNS = 20 )
      PARAMETER ( NRDMAX = 1150 )
      PARAMETER ( NFILES = 10 )
      LOGICAL LIN, NEWSCR, DEFAULT, DUMMY, ASSETS, INITDEP, REDRAW
     + , INITEMP, USRPAT, BORDER
      DIMENSION PAR(NDEPMX,7)
      EQUIVALENCE ( PAR,IPAR )
      REAL LEFT

C
C  PVI ROUTINE TO START GRAPHICS MODE.
C
      CALL JBEGIN
C
C  PVI ROUTINE TO TURN ON SCREEN.
C
      CALL JDINIT(1)
      CALL JDEVON(1)
C
C  PVI ROUTINE TO SAVE FILE FOR PRINTER.
C
      CALL JDINIT(3)
      CALL JDEVON(3)
C
C  PVI ROUTINE TO ENABLE CURSOR CONTROL DEVICE (MOUSE).
C
```

```
      CALL JIENAB(1,2,1)
C
C  INITIALIZE USER PATTERN ARRAY TO FALSE.  (USED BY EMPLOY).
C
      DO 10 I = 1, NWPNS
        NFIL(I) = 0
        USRPAT(I) = .FALSE.
        WPNAME(I) = ' '
   10 CONTINUE
C
C  INITIALIZE NWEAPS ARRAY TO ZEROS.
C
      DO 20 I = 1, NFILES
        NWEAPS(I) = 0
   20 CONTINUE
      NWTOTL = 0
C
C  SET LOGICAL VARIABLES.
C
      INITEMP = .TRUE.
      LIN = .FALSE.
      INITDEP =.TRUE.
      NEWSCR = .FALSE.
      DEFAULT = .FALSE.
      DUMMY = .TRUE.
      ASSETS = .TRUE.
      REDRAW = .FALSE.
      IRECALC = 0
      WXM = 0.0
      WXP = 0.0
      WYM = 0.0
      WYP = 0.0
C
C  INITIALIZE WPNFILE AND NCLR ARRAYS.
C
      DO 30 I = 1, NFILES
        WFILE(I) = 0.0
        NCLR(I) = 1
   30 CONTINUE
      NF = 0
   90 CONTINUE
C
C  THIS SECTION FINDS MAXIMUM AND MINIMUM OF DEPLOYMENT.
C
      IF( NDEPL .LE. 0 ) THEN
      FXM = 0.
      FXP = 1000.0
      FYM = 0.
      FYP = 1000.0
      ELSE
      FXP = XTAR(1,1)
      FXM = XTAR(1,1)
      FYP = XTAR(1,2)
```

```
      FYM = XTAR(1,2)
      DO 40  J = 2,NDEPL
      FXP = AMAX1(FXP,XTAR(J,1))
      FXM = AMIN1(FXM,XTAR(J,1))
      FYP = AMAX1(FYP,XTAR(J,2))
      FYM = AMIN1(FYM,XTAR(J,2))
   40 CONTINUE
      DX = 0.01 * (FXP-FXM)
      DY = 0.01 * (FYP-FYM)
      FXM = FXM - DX
      FXP = FXP + DX
      FYM = FYM - DY
      FYP = FYP + DY
      ENDIF
      IF( LIN )  GO TO 1000
      XXM = FXM
      XXP = FXP
      XYM = FYM
      XYP = FYP
      TXM = FXM
      TXP = FXP
      TYM = FYM
      TYP = FYP
      TXM1 = FXM
      TXP1 = FXP
      TYM1 = FYM
      TYP1 = FYP
      EX = TXP-TXM
      EY = TYP-TYM
C
C  FXX IS FULL DEPLOYMENT MINMAX.
C  TXX IS CURRENT SCREEN SIZE ( INITIAL DEFAULT IS FXX ).
C  XXX IS PREVIOUS SCREEN SIZE.
C  EX AND EY ARE CURRENT SCREEN WIDTH.
C
      CALL REFRESH
  100 CONTINUE
C
C  PRINT MENU SELECTIONS AND PROMPT FOR COMMAND.
C
      PRINT*,'MENU'
      PRINT*,'****'
      PRINT*,'COORDS'
      PRINT*,'DEPLOY'
      PRINT*,'DISTANCE'
      PRINT*,'END'
      PRINT*,'FULL SCALE'
      PRINT*,'GRID'
      PRINT*,'HEADER'
      PRINT*,'HELP "COMMAND"'
      PRINT*,'IDENTIFY'
      PRINT*,'LOCATE'
      PRINT*,'MENU'
```

```
      PRINT*,'MOVE OR REMOVE'
      PRINT*,'REFRESH'
      PRINT*,'SCREEN'
      PRINT*,'SEARCH FOR STRING'
      PRINT*,'SET SCREEN'
      PRINT*,'SYMBOL/COLOR'
      PRINT*,'WEAPON'
      PRINT*,'ZOOM'
      PRINT*,'X - REPEAT CMD'
 1000 CONTINUE
      LIN = .TRUE.
      PRINT*,'COMMAND > '
      READ(*,9)COMAND
    9 FORMAT(A18)
C
C  CHECK WHICH COMMAND INPUT AND CALL APPROPRIATE SUBROUTINES.
C
      IF( COMAND(1:1) .EQ. 'X') COMAND = REPEAT
      REPEAT = COMAND
      IF( COMAND(:3) .EQ. 'IDE') THEN
        CALL IDENTFY(NMV,1)
      ELSE IF ( COMAND(:3) .EQ. 'SYM') THEN
        CALL SYMCOL
      ELSE IF ( COMAND(:3) .EQ. 'DEP') THEN
        CALL DEPLOYA
      ELSE IF ( COMAND(:3) .EQ. 'ZOO') THEN
        CALL ZOOM
      ELSE IF ( COMAND(:3) .EQ. 'REF') THEN
        CALL REFRESH
      ELSE IF ( COMAND(:3) .EQ. 'END') THEN
        PRINT*,'FINISHED ?  (Y OR N) >'
        READ(*,19)ANS
   19   FORMAT(A1)
        IF (ANS .NE. 'Y') THEN
          PRINT*,'RETURN TO AURATEK'
        ELSE
          CALL ENDTEK
          GO TO 9999
        ENDIF
      ELSE IF ( COMAND(:3) .EQ. 'MEN') THEN
        REDRAW = .TRUE.
      ELSE IF ( COMAND(:3) .EQ. 'SET') THEN
        CALL SETSCR
      ELSE IF ( COMAND(:3) .EQ. 'WEA') THEN
        CALL WPNMOD
        INITEMP = .TRUE.
      ELSE IF (COMAND(:3) .EQ. 'HEA') THEN
        CALL HEADER
      ELSE IF ( COMAND(:3) .EQ. 'COO') THEN
        CALL COORDS(*100)
      ELSE IF ( COMAND(:3) .EQ. 'FUL') THEN
        CALL FULLSC
      ELSE IF ( COMAND(:3) .EQ. 'LOC') THEN
```

```
        CALL LOCAT
      ELSE IF ( COMAND(:3) .EQ. 'SCR') THEN
        CALL SCREEN
      ELSE IF ( COMAND(:3) .EQ. 'MOV') THEN
        CALL MOVSUB(1,IRECALC)
      ELSE IF ( COMAND(:3) .EQ. 'REM') THEN
        CALL MOVSUB(2,IRECALC)
      ELSE IF ( COMAND(:3) .EQ. 'DIS') THEN
        CALL DISTANC
        CALL JCLOSE
      ELSE IF ( COMAND(:3) .EQ. 'INV') THEN
        CALL INVIS
      ELSE IF ( COMAND(:3) .EQ. 'RES') THEN
        CALL RESCALE
       ELSE IF ( COMAND(:3) .EQ. 'PRE') THEN
        CALL PREVIUS
      ELSE IF ( COMAND(:3) .EQ. 'HEL') THEN
        CALL HELP(COMAND(6:8))
      ELSE IF ( COMAND(:3) .EQ. 'GRI') THEN
        CALL GRID(TXM,TXP,TYM,TYP)
      ELSE IF ( COMAND(:3) .EQ. 'SEA') THEN
        CALL SEARCH
      ELSE
       PRINT*,'NOT VALID COMMAND'
      ENDIF
C
C  CHECKS TO SEE IF MOVSUB HAS BEEN CALLED.
C
      IF (IRECALC .EQ. 1) THEN
        IRECALC = 0
        GO TO 90
      ENDIF
      IF (REDRAW) THEN
        REDRAW = .FALSE.
        GO TO 100
      ELSE
        GO TO 1000
      ENDIF
 9999 CONTINUE
      RETURN
      END
```

SUBROUTINE WPNIN

FUNCTION: Subroutine WPNIN is used to input weapon data from an AURA "DUMP9" tape.

PARAMETERS: None

DISCUSSION: Subroutine WPNIN is used to input weapon data from an AURA "DUMP9" tape. It reads in the weapon data for x and y burst points for all replications, times and weapon types or for certain types. It then reads in the data one hundred lines at a time.

WPNIN VARIABLE GLOSSARY:

Variable	Definition	Type	Units
FNAME	File name of weapon input.	Char.	N/A
ICH	Flags to clear previous weapon storage.	Char.	N/A
ILINE	Holds input for weapon options.	Char.	N/A
ITEMP	Holds the rest of ILINE.	Char.	N/A
IWIN	Unit number for the file name.	Integer	N/A
IWPN	Weapon number from ILINE.	Integer	N/A
LAREA	Flag if whole weapon area is to be used or selected area.	Logical	N/A
LREP	Flag whether to read all replications. of weapon input or certain replications.	Logical	N/A
LTIM	Flag to use all times or input start and stop time.	Logical	N/A
LWPN	Flag to use all weapon names or certain weapon names.	Logical	N/A
KF	Holds value of IWIN.	Integer	N/A
KWIN	Holds value of WIND.	Real	N/A
KRDS	Keeps count of the rounds.	Real	N/A
KREP	Holds value of MREP.	Real	N/A
MREP	AURA replication for that weapon.	Integer	N/A
MWPN	Number of weapons at that point.	Integer	N/A
NEWINPT	Flag if new weapon input file is called.	Logical	N/A
NF	Counts the number of weapon files used.	Integer	N/A
NTIL	Holds value of IWIN.	Integer	N/A

NFILES	Maximum number of weapon files to be read in at one time.	Integer	N/A
NFTEMP	Number of currently used weapon files.	Integer	N/A
OLDFIL	Flag if old weapon file is be used.	Logical	N/A
OLDNAM	Flag if previous file is being used again.	Logical	N/A
REPLS	Flags for all replications.	Logical	N/A
TIMK	Holds value of TIMM.	Real	sec.
TIMM	Time read in from input file.	Real	sec.
TMPNAM	Weapon file name.	Char.	N/A
WFILE	Real value of integer IWIN.	Real	N/A
WIND	Wind value from input file.	Real	N/A
WPNAM	Name of weapon being read in.	Char.	N/A
WPNAME	Name of weapon being read in.	Char.	N/A
WXMAX	Weapon point X maximum value.	Real	Meters
WXMIN	Weapon point X minimum value.	Real	Meters
WYMAX	Weapon point Y maximum value.	Real	Meters
WYMIN	Weapon point Y minimum value.	Real	Meters
XMA	X, Y & Z coordinates for function point.	Real	Meters
XMD	X, Y & Z coordinates for aim point.	Real	Meters
XKA	Holds value of XMA.	Real	Meters
XKD	Holds value of XMD.	Real	Meters

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: NONE

System Library functions: FLOAT, IFIX, INDEX

Subroutines calling WPNIN: WPNMOD

```
C**********************************WPNIN*****************************************
      SUBROUTINE WPNIN
C
C  WPNIN IS CALLED BY WPNMOD AND IS USED TO INPUT WEAPON
C  DATA FROM OF AN AURA "DUMP9" TAPE.  INPUT MUST OFTEN BE
C  READ IN PARTS SINCE DATA CAN BE VOLUMINOUS.
C
      COMMON/REPS/ REPLS
      COMMON/WPNFILE/ WFILE(NFILES), NCLR(NFILES), NF
      COMMON/FILES/ FNAME(NFILES)
      COMMON/FLAG/ REDRAW
      COMMON/WPNDAT/KREP(NRDMAX),KWPN(NRDMAX),TIMK(NRDMAX),KF(NRDMAX),
     + XKA(NRDMAX,3),XKD(NRDMAX,3),KWIN(NRDMAX),KRDS(NFILES),NFIL(NWPNS)
     + ,NWEAPS(NFILES), IREP(NWPNS), NOWPN(NWPNS), WXMIN(NWPNS), NWPN,
     + WXMAX(NWPNS), WYMIN(NWPNS), WYMAX(NWPNS), WIND(NRDMAX),NWTOTL
      COMMON/WPNCHR/ IWPN(NWPNS), WPNAME(NWPNS)
      CHARACTER*18 ITEMP,ILINE*36, WPNAME, IWPN, ICH*1, WPNAM,
     + FNAME*7, TMPNAM*7
      PARAMETER ( NWPNS = 20 )
      PARAMETER ( NRDMAX = 1150 )
      PARAMETER ( NFILES = 10 )
      LOGICAL LREP,LWPN,LTIM,LAREA,REDRAW,NEWINPT,OLDFIL,OLDNAM,REPLS
      DIMENSION MREP(100),MWPN(100),TIMM(100),XMA(100,3),XMD(100,3)
      REAL KWIN
      NEWINPT = .TRUE.
      OLDNAM = .FALSE.
      OLDFIL = .FALSE.
      REPLS = .FALSE.
C
C  INPUT WEAPON FILE NAME TO BE READ.
C
   20 PRINT*,'CLEAR STORAGE - Y OR N?  >'
      READ(*,9)ICH
    9 FORMAT(A1)
      IF ( ICH .EQ. 'Y' ) THEN
        DO 10 I = 1, NFILES
          KRDS(I) = 0
          WFILE(I) = 0.0
   10   CONTINUE
        NF = 0
        NWTOTL = 0
      ENDIF
   25 PRINT*,'ENTER NAME OF LOCAL WEAPON FILE >'
      READ(*,19)TMPNAM
   19 FORMAT(A7)
C
C  CHECK TO SEE IF IT IS THE OLD WEAPON FILE
C  OR IF A NEW FILE INPUT READ IN.
C
      DO 20 I = 1, NF
        IF (TMPNAM .EQ. FNAME(I)) THEN
          OLDNAM = .TRUE.
          INDX = I
```

```
      ENDIF
   20 CONTINUE
      IF (OLDNAM) THEN
        IWIN = WFILE(INDX)
      ELSE
        IWIN = 11 + NF
        OPEN(IWIN,FILE=TMPNAM)
        FNAME(NF+1) = TMPNAM
      ENDIF
      REWIND IWIN
      DO 30 I = 1, NFILES
        IF (IWIN .EQ. IFIX(WFILE(I))) THEN
          NEWINPT = .FALSE.
          NFTEMP = I
        ENDIF
   30 CONTINUE
      NF = NF + 1
      IF (NEWINPT) THEN
        IF (ICH .EQ. 'N') KRDS(NF) = KRDS(NF-1)
        WFILE(NF) = FLOAT(IWIN)
        NCLR(NF) = NF + 1
      ELSE
        NCLR(NF) = NCLR(NFTEMP)
        KRDS(NF) = KRDS(NF-1)
      ENDIF
      IF ( NCLR(NF) .GT. 7 ) NCLR(NF) = 1
   29 FORMAT( BN, I4 )
      REWIND IWIN
C
C  THE FOLLOWING SPACES OVER THREE BEGINNING LINES FROM AURA.
C
      DO 40 I = 1, 3
      READ(IWIN,39,END=7733)
   39 FORMAT( A1 )
   40 CONTINUE
      DO 1251 I = 1, NWPNS
        READ(IWIN,49,END=7735)NWTEMP,WXMN,WXMX,WYMN,
     +  WYMX, WPNAM
        IF (NWTEMP .EQ. -1) GO TO 1252
   49   FORMAT(I10,20X,4(Ξ10.3),A18)
        IPLACE = INDEX(WPNAM,',')
        IF (IPLACE .GT. 0) THEN
          WPNAM = WPNAM(:IPLACE-1)
        ENDIF
        DO 50 KK = 1, NWTOTL
          IF (IWIN .EQ. NFIL(KK) .AND. WPNAM .EQ. WPNAME(KK)) THEN
            OLDFIL = .TRUE.
          ENDIF
   50   CONTINUE
        IF ( .NOT. OLDFIL ) THEN
          NWTOTL = NWTOTL + 1
          IF (NWTOTL .EQ. NWPNS) THEN
C
```

```
C  MAXIMUM NUMBER OF WEAPON TYPES IS TWENTY.
C
            PRINT*,'TOTAL NO. OF WEAPON TYPES EXCEEDS LIMIT!'
            PRINT*,'>>>>>  CLEAR STORAGE TO CONTINUE  <<<<<'
            NWTOTL = NWTOTL - 1
            GO TO 20
          ENDIF
          NFIL(NWTOTL) = IWIN
          NOWPN(NWTOTL) = NWTEMP
          WXMIN(NWTOTL) = WXMN
          WXMAX(NWTOTL) = WXMX
          WYMIN(NWTOTL) = WYMN
          WYMAX(NWTOTL) = WYMX
          WPNAME(NWTOTL) = WPNAM
          NWEAPS(NF) = 1
        ENDIF
        OLDFIL = .FALSE.
 1251 CONTINUE
 1252 CONTINUE
      ILINE = ' '
C
C  PROMPT TO READ IN ALL REPLICATIONS.
C
      PRINT*,'REPLICATIONS OR ALL? >'
      READ(*,59)ILINE
   59 FORMAT(A1)
      LREP = .FALSE.
      IF( ILINE(1:1) .NE. 'A' )  GO TO 1210
      LREP = .TRUE.
      REPLS = .TRUE.
      GO TO 1220
 1210 NREP = 1
 1211 N1 = INDEX(ILINE,',')
      IF( N1 .EQ. 0 )  GO TO 1214
      REPLS = .TRUE.
      ITEMP = ILINE(:N1-1)
      ILINE = ILINE(N1+1:)
      READ(ITEMP,1209,ERR=7734) IREP(NREP)
      NREP = NREP+1
      IF(NREP .LE. 20 )  GO TO 1211
      NREP = 20
      GO TO 1220
 1214 ITEMP = ILINE
      READ(ITEMP,1209,ERR=7734)  IREP(NREP)
 1220 CONTINUE
      ILINE = ' '
C
C  PROMPT FOR WHICH WEAPON NAMES TO READ IN.
C
      PRINT*,'WEAPON NAME(S) OR ALL? >'
      READ(*,69)ILINE
   69 FORMAT(A34)
      LWPN = .FALSE.
```

```
      IF( ILINE(1:1) .NE. 'A' )  GO TO 1222
      LWPN = .TRUE.
      GO TO 1230
 1222 NWPN = 1
 1221 N1 = INDEX(ILINE,',')
      IF( N1 .EQ. 0 )  GO TO 1224
      ITEMP = ILINE(:N1-1)
      ILINE = ILINE(N1+1:)
      READ(ITEMP,1209,ERR=7734) IWPN(NWPN)
      NWPN = NWPN+1
      IF(NWPN .LE. 20 )  GO TO 1221
      NWPN = 20
      GO TO 1230
 1224 ITEMP = ILINE
      READ(ITEMP,79,ERR=7734)  IWPN(NWPN)
   79 FORMAT( A18 )
 1230 CONTINUE
      ILINE = ' '
C
C  PROMPT FOR START AND STOP TIMES.
C
      PRINT*,'TIMES (START,STOP) OR ALL? >'
      READ(*,89)ILINE
   89 FORMAT(A36)
      LTIM = .FALSE.
      IF( ILINE(1:1) .NE. 'A' )  GO TO 1233
      LTIM = .TRUE.
      GO TO 1240
 1233 CONTINUE
      N1 = INDEX(ILINE,',')
      IF( N1 .EQ. 0 )  GO TO 7734
      ITEMP = ILINE(:N1-1)
      ILINE = ILINE(N1+1:)
      READ(ITEMP,99,ERR=7734) TT1
   99 FORMAT( F10.3 )
      ITEMP = ILINE
      READ(ITEMP,99,ERR=7734)  TT2
 1240 CONTINUE
      ILINE = ' '
C
C  PROMPT FOR THE CORNERS OF SPECIFIC SCREEN DIMENSIONS OR
C  MAXIMIZE THE SCREEN TO FIT ALL OF THE WEAPON LAYDOWN.
C
      PRINT*,'XMIN,XMAX,YMIN,YMAX OR ALL? >'
      READ(*,109)ILINE
  109 FORMAT(A36)
      LAREA = .FALSE.
      IF( ILINE(1:1) .NE. 'A' )  GO TO 1243
      LAREA = .TRUE.
      GO TO 1249
 1243 CONTINUE
      N1 = INDEX(ILINE,',')
      IF( N1 .LE. 0 )  GO TO 7734
```

```
      ITEMP = ILINE(:N1-1)
      READ(ITEMP,119,ERR=7734) TKXM
      ILINE = ILINE(N1+1:)
  119 FORMAT( F10.3 )
      N1 = INDEX(ILINE,',')
      IF( N1 .LE. 0 )  GO TO 7734
      ITEMP = ILINE(:N1-1)
      READ(ITEMP,119,ERR=7734) TKXP
      ILINE = ILINE(N1+1:)
      N1 = INDEX(ILINE,',')
      IF( N1 .LE. 0 )  GO TO 7734
      ITEMP = ILINE(:N1-1)
      READ(ITEMP,119,ERR=7734) TKYM
      ILINE = ILINE(N1+1:)
      READ(ILINE,119,ERR=7734) TKYP
 1249 CONTINUE
C
C  NOW READ IN, BY THE BUFFER LOAD, AND CULL. THIS SECTION FIRST
C  LOOKS FOR THE NUMBER OF LINES TO READ IN THEN READS IN THAT
C  NUMBER OF LINES.  IT DOES THIS UNTIL ALL DATA HAS BEEN READ IN.
C
 1250 CONTINUE
      READ(IWIN,129,END=1295) MRDS,(MREP(I),MWPN(I),TIMM(I),
     +(XMA(I,J),J=1,3),(XMD(I,J),J=1,3),WIND(I),I=1,MRDS)
  129 FORMAT( I10,/,100(2I10,8F10.3,/) )
      DO 60 I = 1, MRDS
        IF(LREP)  GO TO 1260
        DO 1255  J = 1, NREP
        IF( MREP(I) .EQ. IREP(J) )  GO TO 1260
   60   CONTINUE
        GO TO 1290
 1260   IF( LWPN )  GO TO 1270
        DO 70  J = 1, NWPNS
        IF( MWPN(I) .EQ. NOWPN(J) )  GO TO 1270
   70   CONTINUE
        GO TO 1290
C
C  CHECKS TIME READ IN.
C
 1270   IF( LTIM )  GO TO 1272
        IF( TIMM(I) .LT. TT1 .OR. TIMM(I) .GT. TT2 )  GO TO 1290
C
C  CHECKS AREA READ IN.
C
 1272   IF(LAREA)  GO TO 1280
        IF( XMA(I,1) .LT. TKXM .OR. XMA(I,1) .GT. TKXP )  GO TO 1290
        IF( XMA(I,2) .LT. TKYM .OR. XMA(I,2) .GT. TKYP )  GO TO 1290
 1280   KRDS(NF) = KRDS(NF)+1
C
C  CHECKS NUMBER OF ROUNDS READ IN.
C
        IF( KRDS(NF) .LE. NRDMAX )  GO TO 1281
        KRDS(NF) = NRDMAX
```

```
      PRINT*,'INPUT EXCEEDS NRDMAX'
      PRINT*,'INPUT TRUNCATED'
      GO TO 1295
1281   KREP(KRDS(NF)) = MREP(I)
      KWPN(KRDS(NF)) = MWPN(I)
      KWIN(KRDS(NF)) = WIND(I)
      TIMK(KRDS(NF)) = TIMM(I)
      KF(KRDS(NF)) = IWIN
      DO 80 J = 1,3
      XKA(KRDS(NF),J) = XMA(I,J)
 80   XKD(KRDS(NF),J) = XMD(I,J)
1290 CONTINUE
   GO TO 1250
1295 CONTINUE
   RETURN
7733 CONTINUE
   PRINT*,'NO DATA ON INPUT FILE!'
   PRINT*,'** RETURN TO WEAPON MODE **'
   RETURN
7734 CONTINUE
   PRINT*,'INPUT ERROR. GO BACK TO'
   RETURN
7735 CONTINUE
   PRINT*,'END OF FILE ENCOUNTERED ON WEAPON INPUT!!'
   PRINT*,'TRY AGAIN!'
   RETURN
   END
```

SUBROUTINE WPNMOD

FUNCTION: Subroutine WPNMOD is used to display the menu for implementing weapons options.

PARAMETERS: None

DISCUSSION: Subroutine WPNMOD is used to display the menu for implementing weapons options. WPNMOD prompts the user for command then executes the proper subroutine for that option.

WPNMOD VARIABLE GLOSSARY:

Variable	Definition	Type	Units
EX	Difference between X maximum and X minimum values found in ZOOM.	Real	N/A
EY	Difference between Y maximum and Y minimum values found in ZOOM.	Real	N/A
NEWSCR	Flags if a new screen needs to be drawn.	Logical	N/A
REDRAW	Flag to see if screen needs to be redrawn.	Logical	N/A
REPEAT	Holds last command used.	Char.	N/A
TXM	Minimum value of XZ(1) and XZ(2) found in ZOOM.	Real	Meters
TXP	Maximum value of XZ(1) and XZ(2) found in ZOOM.	Real	Meters
TYM	Minimum value of YZ(1) and YZ(2) found in ZOOM.	Real	Meters
TYP	Maximum value of YZ(1) and YZ(2) found in ZOOM.	Real	Meters

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: COORDS DISTANC EMPLOY FULLSC GRID HEADER HELP IDENTFY LIST LOCAT REFRESH SCREEN USERPAT WPNIN ZOOM

System Library functions: NONE

Subroutines calling WPNMOD: TARMOD

```
C*************************WPNMOD*****************************************
      SUBROUTINE WPNMOD
C
C  WEAPON MODE IS THE MAIN GRAPHICS PROGRAM FOR IMPLEMENTING WEAPON
C  OPTIONS.
C
      COMMON/BRDR/ BORDER
      COMMON/WPNFILE/ WFILE(NFILES), NCLR(NFILES), NF
      COMMON/FLAG/ REDRAW
      COMMON/WDIM/ WXM, WXP, WYM, WYP
      COMMON/PAD/TXM,TXP,TYM,TYP, LEFT, FACTOR, TOP,FXM,FXP,FYM,FYP
      COMMON/WIDTH/ EX, EY
      COMMON/SCRDAT/ XXM, XXP, XYM, XYP, TXM1, TXP1, TYM1, TYP1
      COMMON/SCRFLG/ INITEMP
      CHARACTER*18 COMAND, ANS*1, REPEAT
      PARAMETER ( NFILES = 10 )
      LOGICAL LIN, REDRAW, NEWSCR, INITEMP
      DATA LIN /.FALSE./
      REAL LEFT

      NEWSCR = .TRUE.
  100 CONTINUE
C
C  DISPLAY WEAPON MENU ON SCREEN.
C
      PRINT*,'WEAPON MENU'
      PRINT*,'***********'
      PRINT*,'BACK'
      PRINT*,'CLEAR PATTERNS'
      PRINT*,'COORDS'
      PRINT*,'DISTANCE'
      PRINT*,'EMPLOY'
      PRINT*,'FULL SCALE'
      PRINT*,'GRID'
      PRINT*,'HEADER'
      PRINT*,'HELP "COMMAND"'
      PRINT*,'IDENTIFY'
      PRINT*,'INPUT'
      PRINT*,'LIST'
      PRINT*,'LOCATE'
      PRINT*,'MENU'
      PRINT*,'REFRESH'
      PRINT*,'ZOOM'
      PRINT*,'SCREEN'
      PRINT*,'X - REPEAT CMD'
 1000 CONTINUE
      LIN = .TRUE.
C
C  SETS IF NEWSCREEN NEEDS TO BE DRAWN.
C
     IF (NEWSCR) THEN
       NEWSCR = .FALSE.
     ENDIF
```

```
      PRINT*,'COMMAND(WPN MODE) > '
      READ(*,9)COMAND
    9 FORMAT(A12)
C
C  REPEATS LAST COMMAND.
C
      IF( COMAND(1:1) .EQ. 'X' )  COMAND = REPEAT
      REPEAT = COMAND
      IF(COMAND(:3) .EQ. 'IDE') THEN
          PRINT*,'TARGET OR WEAPON >'
          READ(*,19)ANS
   19     FORMAT(A1)
          IF (ANS .EQ. 'T') THEN
            CALL IDENTFY(NMV,1)
          ELSE
           CALL IDENTFY(NMV,2)
          ENDIF
      ELSE IF(COMAND(:3) .EQ. 'INP' ) THEN
        CALL WPNIN
        NEWSCR = .TRUE.
      ELSE IF (COMAND(:3) .EQ. 'PAT') THEN
        CALL USERPAT
      ELSE IF ((COMAND(:3) .EQ. 'BAC').OR.(COMAND(:3) .EQ. 'END')) THEN
        CALL REFRESH
        REDRAW = .TRUE.
        RETURN
      ELSE IF (COMAND(:3) .EQ. 'MEN' ) THEN
        NEWSCR = .TRUE.
        REDRAW = .TRUE.
      ELSE IF (COMAND(:3) .EQ. 'COO') THEN
        CALL COORDS
      ELSE IF( COMAND(:3) .EQ. 'EMP' )  THEN
        CALL EMPLOY
        NEWSCR = .TRUE.
      ELSE IF ( COMAND(:3) .EQ. 'FUL' ) THEN
        CALL FULLSC
      ELSE IF ( COMAND(:3) .EQ. 'LIS' ) THEN
        CALL LIST
      ELSE IF( COMAND(:3) .EQ. 'LOC' )  THEN
        CALL LOCAT
      ELSE IF( COMAND(:3) .EQ. 'DIS' )  THEN
        CALL DISTANC
        CALL JCLOSE
      ELSE IF ( COMAND(:3) .EQ. 'REF' ) THEN
        CALL REFRESH
        NEWSCR = .TRUE.
      ELSE IF (COMAND(:3) .EQ. 'CLE') THEN
        CALL REFRESH
        NEWSCR = .TRUE.
      ELSE IF (COMAND(:3) .EQ. 'HEA') THEN
        CALL HEADER
      ELSE IF (COMAND(:3) .EQ. 'ZOO') THEN
        CALL ZOOM
```

```
      ELSE IF ( COMAND(:3) .EQ. 'SCR' ) THEN
        CALL SCREEN
      ELSE IF (COMAND(:3) .EQ. 'GRI') THEN
        CALL GRID(TXM,TXP,TYM,TYP)
      ELSE IF (COMAND(:3) .EQ. 'HEL') THEN
        CALL HELP(COMAND(6:8))
      ELSE
        PRINT*,'NOT VALID COMMAND'
      ENDIF
C
C  IF REDRAW IS TRUE, RESET TO FALSE,
C  REPEAT MENU AND PROMPT FOR NEW COMMAND
C  ELSE JUST PROMPT FOR NEXT COMMAND.
C
      IF (REDRAW) THEN
        REDRAW = .FALSE.
        GO TO 100
      ELSE
        GO TO 1000
      ENDIF
      END
```

SUBROUTINE ZOOM

FUNCTION: Subroutine ZOOM allows the user to designate a region of the deployment to be enlarged in size and the screen reset to the new size.

PARAMETERS: None

DISCUSSION: Subroutine ZOOM allows the user to designate a region of the deployment to be enlarged in size and the screen reset to the new size. REFRESH is called to redraw the screen at the enlarged dimensions.

ZOOM VARIABLE GLOSSARY:

Variable	Definition	Type	Units
EX	Difference between X maximum and X minimum values.	Real	Meters
EY	Difference between Y maximum and Y minimum values.	Real	Meters
TXM	Minimum value of XZ(1) and XZ(2).	Real	Meters
TXP	Maximum value of XZ(1) and XZ(2).	Real	Meters
TYM	Minimum value of YZ(1) and YZ(2).	Real	Meters
TYP	Maximum value of YZ(1) and YZ(2).	Real	Meters
XZ	X world coordinate used in PVI calls.	Real	Meters
XXM	Holds value TXM.	Real	Meters
XXP	Holds value TXP.	Real	Meters
XYM	Holds value TYM.	Real	Meters
XYP	Holds value TYP.	Real	Meters
YZ	Y world coordinate used in PVI calls.	Real	Meters

SUBROUTINE CROSS REFERENCE TABLE:

Program subroutines called: REFRESH

System Library functions: AMAX1, AMIN1

Subroutines calling ZOOM: TARMOD WPNMOD

```
C**************************ZOOM*********************************************
      SUBROUTINE ZOOM
C
C  ZOOM ALLOWS THE USER TO DESIGNATE A REGION OF THE DEPLOYMENT
C  WHICH IS THEN ENLARGED IN SIZE AND THE SCREEN IS RESET.
C
      COMMON/WIDTH/ EX, EY
      COMMON/SCRDAT/ XXM, XXP, XYM, XYP, TXM1, TXP1, TYM1, TYP1
      COMMON/FLAG/ REDRAW
      COMMON/PAD/TXM,TXP,TYM,TYP, LEFT, FACTOR, TOP,FXM,FXP,FYM,FYP
      LOGICAL  REDRAW
      DIMENSION XZ(2), YZ(2)
C
C  X LIMITS ARE THE MOST RECENT PAST LIMITS ( FOR PREVIOUS OPTION).
C  T LIMITS ARE CURRENT LIMITS.
C  U AND V LIMITS ARE SUBSETS ( FOR BULK OPTION ).
C
      XXM = TXM
      XXP = TXP
      XYM = TYM
      XYP = TYP
C
C  PVI ROUTINE TO OPEN A TEMPORARY SEGMENT
C  AND SET THE COLOR
C
      CALL JOPEN
      CALL JCOLOR(3)
C
C  LOOP TO FIND THE TWO CORNERS TO ZOOM IN ON
C
      DO 10  J = 1, 2
C
C  PVI ROUTINE TO ENABLE MOUSE/INPUT DEVICE
C
        CALL JLOCAT(1,1,1,b,XT,YT)
C
C  PVI ROUTINE CONVERTS LOCATOR OUTPUT FROM VIRTUAL COORDINATES
C  TO WORLD COORDINATES.
C
        CALL JCONVW(XT,YT,XZ(J),YZ(J),Z)
C
C  PVI ROUTINE TO DRAW THE BOX TO ZOOM ON.
C
        CALL JMOVE(TXM,YZ(J))
        CALL JDRAW(TXP,YZ(J))
        CALL JMOVE(XZ(J),TYM)
        CALL JDRAW(XZ(J),TYP)
   10 CONTINUE
      CALL JCOLOR(7)
C
C  SET DIMENSIONS FOR ENLARGED AREA.
C
      TXP = AMAX1(XZ(1),XZ(2))
```

```
      TXM = AMIN1(XZ(1),XZ(2))
      TYP = AMAX1(YZ(1),YZ(2))
      TYM = AMIN1(YZ(1),YZ(2))
      EX = TXP-TXM
      EY = TYP-TYM
C
C  CLOSES A SEGMENT.
C
      CALL JCLOSE
C
C  REFRESH DRAWS THE ENLARGED AREA.
C
      CALL REFRESH
      REDRAW = .TRUE.
      RETURN
      END
```

APPENDIX B

AURATEK Variable Definitions

INTENTIONALLY LEFT BLANK.

Variable	Definition	Type	Units
A	Max. value of XSEMI and YSEMI.	Real	N/A
ALTPOS	Holds alternate posture values.	Char.	N/A
ASSETS	Flag used in INVIS subroutine.	Logical	N/A
AXIS	Greater value of XLEN and YLEN.	Real	Meters
B	Min. value of XSEMI and YSEMI.	Real	N/A
BORDER	Turns border on or off around the work area.	Logical	N/A
C1	Cosine of the value WIND1.	Real	N/A
C2	Cosine of the value P.	Real	N/A
C3	Value of T1.	Real	Radians
CMD	Holds first three characters of command.	Char.	N/A
COMM	Holds commented deployment line.	Char.	N/A
COMAND	Holds value of COMMAND from TARMOD.	Char.	N/A
D	Distance between the two points.	Real	Meters
DEFAULT	Flags if default postures are to be used.	Logical	N/A
DUMMY	Flag used in invisible option.	Logical	N/A
DX	One percent of X length.	Real	Meters
DY	One percent of Y length.	Real	Meters
EX	Difference between X max. and X min. values found in ZOOM	Real	N/A
EY	Difference between Y max. and Y min. values found in ZOOM	Real	N/A
FNAME	File name of weapon input.	Char.	N/A
FXM	Min. value of FXM and XX found in ZOOM.	Real	Meters
FXP	Min. value of FXP and XX found in ZOOM.	Real	Meters
FYM	Min. value of FYM and YY found in ZOOM.	Real	Meters
FYP	Min. value of FYP and YY found in ZOOM.	Real	Meters
H	X coordinate offset for weapon pattern	Real	Radians
IANS	Input of what type of string to change.	Char.	N/A
ICH	Flags to clear previous weapon storage.	Char.	N/A
ICLR	Holds number of selected color.	Char.	N/A
ICOLOR	Sets color value for symbol written to the output file.	Integer	N/A
ID	Holds value of NAME.	Char.	N/A

Variable	Definition	Type	Units
IEND	Holds number of blanks found in ID.	Integer	N/A
ILINE	Reads in each line of deployment.	Char.	N/A
INITDEP	Flags the first time a deployment option is used.	Logical	N/A
IPAR	Array of kill criteria & initial postures.	Integer	N/A
ISTART	Value to start weapon number.	Integer	N/A
ISY	Holds number of selected symbol.	Char.	N/A
ISYM	Sets symbol value written to to the output file.	Integer	N/A
IT	Set to value of ITMP.	Integer	N/A
ITEMP	Holds the rest of ILINE.	Char.	N/A
ITMP	Set to the number of deployment points.	Integer	N/A
IVIS	Character denoting invisible point.	Char.	N/A
IWIN	Unit number for the file name.	Integer	N/A
IWORD	Holds first word of deployment line to compare.	Char.	N/A
IWPN	Weapon number from ILINE	Integer	N/A
K	Y coordinate offset for weapon pattern.	Real	Meters
KF	Set to value of IWIN.	Integer	N/A
KRDS	Number of rounds from weapon file.	Integer	N/A
KREP	Number of replications from file.	Integer	N/A
KWIN	Set to the value of WIND	Real	N/A
KWPN	Set to the value of MWPN	Integer	N/A
L7	Used to flag end of data in ILINE.	Logical	N/A
LAREA	Flag if whole weapon area is to be used or selected area.	Logical	N/A
LCENT	Flag to mark weapon centers.	Logical	N/A
LREP	Flag whether to read all replications of weapon input or certain replications.	Logical	N/A
LTIM	Flag to use all times or input start and stop time.	Logical	N/A
LWPN	Flag to use all weapon names or certain weapon names.	Logical	N/A
MOPP	Used to flag "MOPP ALL" line in the input file.	Logical	N/A
MOPPALL	Equals ILINE if 'MOPP ALL' is in	Char.	N/A
MREP	AURA replication for that weapon.	Integer	N/A
MWPN	Number of weapons at that point.	Integer	N/A
N1	Keeps count of commas separating target target description.	Integer	N/A

Variable	Definition	Type	Units
NAME	Asset name.	Char.	N/A
NCLR	Number of weapon files read in.	Integer	N/A
NDEPL	Number of deployment points.	Integer	N/A
NEWINPT	Flag if new weapon input file is called.	Logical	N/A
NEWSCR	Flags if a new screen needs to be drawn.	Logical	N/A
NF	Counts the number of weapon files used.	Integer	N/A
NFIL	Holds value of IWIN.	Integer	N/A
NFILA	Tape number where weapon file is written.	Integer	N/A
NFILES	Max. number of weapon files to be read in at one time.	Integer	N/A
NFTEMP	Number of currently used weapon files.	Integer	N/A
NIN	Counter to rearrange assets.	Integer	N/A
NMV	Number of assets at a deployment point.	Integer	N/A
NOUT	Counter for ITMP values.	Integer	N/A
NOWPN	Set to the value of NWTEMP.	Integer	N/A
NPTS	Number of points to draw the ellipse.	Integer	N/A
NVIS	Number of symbols visible.	Integer	N/A
NWPNS	Total number of weapon types.	Integer	N/A
NWTOTL	Total number of weapons read in.	Integer	N/A
OFFSETS	Holds offset value of deployment used if certain groups of assets are duplicated within the deployment file.	Real	N/A
OLDFIL	Flags if weapon file has been read before.	Logical	N/A
OLDNAM	Flag if previous file is being used again.	Logical	N/A
P	Increment angle for ellipse	Real	Radians
PAR	Array holding number of assets with the same name at that point.	Real	N/A
PASS1	Flags for all replications of weapon.	Logical	N/A
PRT	Real of integer IPRT.	Real	N/A
REDRAW	Flag to see if screen needs to be redrawn.	Logical	N/A
REPEAT	Holds last command used.	Char.	N/A
REPLS	Flags to ask employ question.	Logical	N/A
S1	Sine of the value WIND1.	Real	N/A
S2	Sine of the value P.	Real	N/A
S3	New increment value for sine angle.	Real	Radians

Variable	Definition	Type	Units
SET	Used to set different size letters.	Real	N/A
SCRMAX	Maximum length with weapons deployed.	Real	N/A
T1	Calculates new increment for cosine angle.	Real	radians
TIC	One tenth the value of AXIS.	Real	Meters
TIMK	Time read in from weapon input file.	Real	N/A
TIMM	Holds value of TIMK.	Real	sec.
TMOVE	Set to the value of TIC.	Real	Meters
TMPNAM	Weapon file name.	Char.	N/A
TT	One half of TY or TX.	Real	Meters
TX	Max. value X minus Min. value of X.	Real	Meters
TXM	Min. value of XZ(1) and XZ(2).	Real	Meters
TXM1	Set to value of FXM.	Real	Meters
TXMID	Midpoint of X length.	Real	Meters
TXP	Max. value of XZ(1) and XZ(2).	Real	Meters
TXP1	Max. X value used in DI-3000's jwindo.	Real	Meters
TXXM	Maximum screen coordinate in X direction.	Real	N/A
TXXP	Minimum screen coordinate in X direction.	Real	N/A
TY	Max. value Y minus Min. value of Y.	Real	Meters
TYM	Min. value of YZ(1) and YZ(2).	Real	Meters
TYM1	Min. Y value used in DI-3000's jwindo.	Real	Meters
TYMID	Midpoint of Y length.	Real	Meters
TYYM	Maximum screen coordinate in Y direction.	Real	N/A
TYP	Max. value of YZ(1) and YZ(2).	Real	Meters
TYP1	Set of value of FYM.	Real	Meters
TYYP	Minimum screen coordinate in Y direction.	Real	N/A
USRPAT	Flags if USRPAT has been called before.	Logical	N/A
UXMAX	Max. X of weapon pattern.	Real	N/A
UXMIN	Min. X of weapon pattern.	Real	N/A
UYMAX	Max. Y of weapon pattern.	Real	N/A
UYMIN	Min. Y of weapon pattern.	Real	N/A
WIND	Wind value from input file.	Real	N/A
WIND1	Value of Wind converted to radians.	Real	N/A
WFILE	Unit number for the file name.	Real	N/A
WPNAM	Name of weapon being read in.	Char.	N/A
WPNAME	Weapon name from weapon input file.	Char.	N/A
WXM	Set to the value of TXM.	Real	N/A

Variable	Definition	Type	Units
WXMAX	Set to the value of WXMX.	Real	N/A
WYM	Max. screen coordinate in Y direction.	Real	N/A
WXMIN	Set to the value of WXMN.	Real	N/A
WYMIN	Weapon point Y min. value.	Real	Meters
WXP	Set to the value of TXP.	Real	N/A
WYM	Set to the value of TYM.	Real	N/A
WYMAX	Set to the value of WYMX.	Real	N/A
WYMIN	Set to the value of WYMN.	Real	N/A
WYP	Set to the value of TYP.	Real	N/A
X1	Long axis of ellipse.	Real	Meters
XDIAM	Half of length in X direction.	Real	N/A
XKA	X and Y coordinate to mark weapon.	Real	N/A
XKD	X, Y & Z coordinates for aim point of weapon.	Real	N/A
XLEN	Length in X direction.	Real	Meters
XMA	X, Y & Z coordinates for function.	Real	Meters
XMD	X, Y & Z coordinates for aim point.	Real	Meters
XPAT	X coordinate used to draw ellipse.	Real	N/A
XSEMI	Half of length in X direction.	Real	N/A
XTAR	Array of x and y coordinates of asset deployment.	Real	N/A
XXM	Holds value TXM.	Real	Meters
XXP	Holds value TXP.	Real	Meters
XYM	Holds value TYM.	Real	Meters
XYP	Holds value TYP.	Real	Meters
XZ	X world coordinate used in PVI calls.	Real	Meters
Y1	Short angle of ellipse.	Real	Meters
YDIAM	Half of length in Y direction.	Real	N/A
YLEN	Length in Y direction.	Real	Meters
YPAT	Y coordinate used to draw ellipse.	Real	N/A
YSEMI	Half of length in Y direction.	Real	N/A
YSIZE	Used to scale letters to layout.	Real	N/A
YZ	Y world coordinate used in PVI calls.	Real	Meters

INTENTIONALLY LEFT BLANK.

No of Copies	Organization
2	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22304-6145
1	HQDA (SARD-TR) WASH DC 20310-0001
1	Commander US Army Materiel Command ATTN: AMCDRA-ST 5001 Eisenhower Avenue Alexandria, VA 22333-0001
1	Commander US Army Laboratory Command ATTN: AMSLC-DL Adelphi, MD 20783-1145
2	Commander US Army, ARDEC ATTN: SMCAR-IMI-I Picatinny Arsenal, NJ 07806-5000
2	Commander US Army, ARDEC ATTN: SMCAR-TDC Picatinny Arsenal, NJ 07806-5000
1	Director Benet Weapons Laboratory US Army, ARDEC ATTN: SMCAR-CCB-TL Watervliet, NY 12189-4050
1	Commander US Army Armament, Munitions and Chemical Command ATTN: SMCAR-ESP-L Rock Island, IL 61299-5000
1	Director US Army Aviation Research and Technology Activity ATTN: SAVRT-R (Library) M/S 219-3 Ames Research Center Moffett Field, CA 94035-1000

No of Copies	Organization
1	Commander US Army Missile Command ATTN: AMSMI-RD-CS-R (DOC) Redstone Arsenal, AL 35898-5010
1	Commander US Army Tank-Automotive Command ATTN: AMSTA-TSL (Technical Library) Warren, MI 48397-5000
1	Director US Army TRADOC Analysis Command ATTN: ATRC-WSR White Sands Missile Range, NM 88002-5502
(Class. only) 1	Commandant US Army Infantry School ATTN: ATSH-CD (Security Mgr.) Fort Benning, GA 31905-5660
(Unclass. only) 1	Commandant US Army Infantry School ATTN: ATSH-CD-CSO-OR Fort Benning, GA 31905-5660
1	Air Force Armament Laboratory ATTN: AFATL/DLODL Eglin AFB, FL 32542-5000
	Aberdeen Proving Ground
2	Dir, USAMSAA ATTN: AMXSY-D AMXSY-MP, H. Cohen
1	Cdr, USATECOM ATTN: AMSTE-TD
3	Cdr, CRDEC, AMCCOM ATTN: SMCCR-RSP-A SMCCR-MU SMCCR-MSI
1	Dir, VLAMO ATTN: AMSLC-VL-D

No. of Copies | Organization

1 HQDA
OASA (RDA)
ATTN: SARD-DOS, MAJ Gary R. Davis
WASH DC 20310-0103

10 C.I.A
ATTN: OIR/DB/Standard
GE-47 HQ
Washington, DC 20505

2 Director
Defense Nuclear Agency
ATTN: STBE,
Dr. D. Auton
Dr. R. Young
Washington, DC 20305

2 Commander
Defense Nuclear Agency/Field Command
ATTN: FCPR,
MAJ Fleming
CPT Thorton
Kirtland AFB, NM 87115

3 DA/ODCSOPS
Pentagon
ATTN: DAMO-ZDF, Mr. Wolpert
DAMO-ZXG, Mr. Patterson
DAMO-ZDF
Washington, DC 20310-0404

1 Commander
U.S. Army Materiel Command
ATTN: AMCDRA-AT
5001 Eisenhower Avenue
Alexandria, VA 22333-0001

2 Commander
U.S. Army ARDEC
ATTN: SMCAR-FSS,
Mr. Ostuni
Mr. Brooks
Picatinny Arsenal, NJ 07806-5000

1 CDR
CATA-NSC
ATTN: ATZL-TAC-DC, CPT H. Lee
Ft. Leavenworth, KS 66027

No. of Copies | Organization

1 Commander
HQ AMCCOM
ATTN: AMSMC-IMP-IL
Rock Island, IL 61299-7300

1 Commander, USACECOM
R&D Technical Library
ATTN: ASQNC-ELC-I-T, Myer Center
Fort Monmouth, NJ 07703-5301

5 Commander
U.S. Army Harry Diamond Laboratories
ATTN: SLCHD-NW-RA
2800 Powder Mill Road
Adelphia, MD 20783

2 Director
U.S. Army Missile and Space Intelligence Center
ATTN: AIAMS-YDL
AIAMS-TSL
Redstone Arsenal, AL 35898-5500

2 Commander
TRAC-WSMR
ATTN: ATRC-WSR, Mr. Belk
ATOR-TSL
White Sands Missile Range, NM 88002-5522

2 Director
TRAC-FLVN
ATTN: ATRC-FTC, CPT Perry
ATRC-F, Dr. Robert LaRocque
Ft. Leavenworth, KS 66027

1 Director
TRAC RPD
ATTN: ATRC-RPR, Mr. Radda
Ft. Monroe, VA 23651-5143

2 Commander
U.S. Army Aviation Logistics Center
ATTN: ATSP-CD,
CPT White
Mr. Pollack
Ft. Eustis, VA 23604

No. of Copies | Organization

3 Commander
U.S. Army Combined Arms Combat Developments Agency
ATTN: ATZL-CAS-SP, CPT DeWulf
ATZL-CAP,
COL Henderson
CPT Grand
Ft. Leavenworth, KS 66027

2 U.S. Army Nuclear and Chemical Agency
ATTN: MONA-WE
MONA-CM, MAJ Joe Fernandez
7500 Backlick Road, Bldg 2073
Springfield, VA 22150-3198

1 Director
TRAC-Ft. Lee
ATTN: ATRC-OS
Ft. Lee, VA 23801-6000

3 Director
Concepts Analysis Agency
ATTN: CSCA,
Mr. D. Hurd
LTC Barrett
Mr. Chandler
8120 Woodmont Avenue
Bethesda, MD 20014

1 Commandant
U.S. Army Air Defense Artillery School
ATTN: ATZK-CD
Bldg 5800
Ft. Bliss, TX 79916

1 Commandant
U.S. Army Armor School
ATTN: ATZK-CP-SD, Mr. Hekemeyer
Ft. Knox, KY 40121

1 Commander
U.S. Army Engineering School
ATTN: ATSE-CDM
Ft. Leonard Wood, MO 65473-6620

1 Commander
U.S. Army Field Artillery School
ATTN: ATSF-CD
Ft. Sill, OK 73503

No. of Copies | Organization

1 Commandant
U.S. Army Transportation School
ATTN: ATSP-PD-C
Ft. Eustis, VA 23604

2 Commander
U.S. Army Missile and Munitions Center and School
ATTN: ATSK-CD
Redstone Arsenal, AL 35898

1 Commander
U.S. Army Quartermaster School
ATTN: ATSM-CDC
Ft. Lee, VA 23801

1 Commander
U.S. Army Signal Center & School
Directorate of Combat Developments
Ft. Gordon, GA 30905

4 Commandant
U.S. Army Chemical School Directorate of Combat Developments
ATTN: ATZN-CM-CC,
LTC Gary Stratton
Mr. B. Gould
Ms. Sherry Barnes
Ms. Rena Seals
Ft. McClellan, AL 36201

1 US Army War College
ATTN: Mr. Carl Barton
Box 531
Carlisle Barracks, PA 17013-5050

1 Commander
Armed Forces Radiobiological Research Institute
National Naval Medical Center
ATTN: Mr. William Jackson
Bethesda, MD 20014

2 Commander
Combined Arms Center
Scores Working Group
ATTN: ATZL-CAD-LN
Ft. Leavenworth, KS 66027

No. of Copies | Organization

2 Commander
Naval Surface Warfare Center
ATTN: Code 31-TJY,
Mr. Yencha
Mr. P. Kirk
Dahlgren, VA 22448

1 President
Center for Naval Analyses
4401 Ford Avenue
P.O. Box 16268
Alexandria, VA 22302-0268

2 Walter Reed Army Institute of Research
Division of Medicine
Department of Respiratory Research
ATTN: SGRD-UWH-E,
MAJ Gary R. Ripple, MD
LTC Kenneth Torrington
Washington, DC 20307-5100

3 Institute for Defense Analysis
ATTN: Mr. Graham McBryde
Mr. M. Kerlin
Dr. M. Charren
1801 N. Beauregard Street
Alexandria, VA 22311

3 U.S. Army Personnel Command
ATTN: DAPC-MOP, H. Rowland Ludden
TAPC-MOP,
MAJ M. Brice Elliot
CPT I. McMahon
200 Stovall Street
Alexandria, VA 22332-0432

1 U.S. Army Chief of Staff
ATTN: DACS-DPZ-A
The Pentagon
Washington, DC 20310

3 U.S. Army Deputy Chief of Staff
for Operations and Plans
ATTN: DAMO-SWC, LTC Gordon Miller
DAMO-ZD,
Mr. John Riente
Mr. James Metzger
The Pentagon
Washington, DC 20310

No. of Copies | Organization

1 Defense Nuclear Agency
6801 Telegraph Road
Alexandria, VA 22310

2 Office of the Surgeon General
5111 Leesburg Pike
ATTN: DASG-HCO-P, MAJ Ray Keith
DASG-HCD, MAJ Robert Eng
Falls Church, VA 22041

1 U.S Army Engineer Studies Center
ATTN: ESC-EO
Casey Bldg 2594
Ft. Belvoir, VA 22060-5583

1 Commanding General
U.S. Army Operational Test and
Evaluation Agency
Park Center IV
4501 Ford Avenue
Alexandria, VA 22302-1428

2 Director
TRAC-MTRY
Naval Postgraduate School
ATTN: LTC Vernon M. Bettencourt, Jr.
MAJ R. Wimberley
P.O. Box 8692
Monterey, CA 93943-0692

1 U.S. Army Command and General
Staff College
Office of the Commandant
Ft. Leavenworth, KS 66027-6900

3 JAYCOR
ATTN: Mr. J. David Claiborne
Mr. Hubert Lockerd
Mr. William Jenkins
P.O. Box 87
Edgewood, MD 21040

3 EAI Corporation
ATTN: Mr. Dennis Metz
Mr. Bruce Fischer
Mr. S. Matthew Hutton
1308 Continental Drive, Suite J
Abingdon, MD 21009

No. of Copies | Organization

2 Science Applications International Corporation
ATTN: R. McNally
M. Stark
626 Towne Center Dr.
Joppa, MD 21085

1 Battelle
ATTN: D.W. Harper
7501 Memorial Parkway South
Suite 101
Huntsville, AL 35802-2258

2 WRDC/FIVS/SURVIAC
Wright-Patterson AFB, OH 45433-6553

No. of Copies | Organization

Aberdeen Proving Ground

6 Dir, USAMSAA
ATTN: AMXSY,
Mr. M. Miller
Ms. C. Horley
Ms. D. Smoot
AMXSY-C, M. Carroll
AMXSY-G, M. Sandmeyer
AMXSY-J, R. Mezan

8 Cdr, CRDEC, AMCCOM
ATTN: AMSTE-SI-F
SMCCR-ST,
Mr. D. Sloop
Mr. R. Jablonski
Mr. J. Razulis
Mr. J. Walther
Mr. C. Crawford
Mr. L. Davis
Mr. L. Baer

2 Cdr, USAHEL
ATTN: W. DeBellis
O. Zubal

1 Cdr, USAOC&S
ATTN: ATSL-CD-CS

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers below will aid us in our efforts.

1. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

2. How, specifically, is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

3. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

4. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

BRL Report Number BRL-MR-3896 Division Symbol _____

Check here if desire to be removed from distribution list. ____

Check here for address change. ____

Current address: Organization _____
Address _____
_____

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS

BUSINESS REPLY MAIL
FIRST CLASS PERMIT No 0001, APG, MD

Postage will be paid by addressee

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES