PARAMETRIC ENGINEERING
SYSTEM DEFINITION MODEL

VOLUME II
APPENDIX C (FORTRAN LISTINGS)

July 1979
Contract DAAK30-78-C-0059

by
S. Spaulding, A. Weintraub,
F. Cioch, J. Lenz

Vector Research, Incorporated
P.O. Box 1506
Ann Arbor, Michigan 48106

U.S. ARMY TANK-AUTOMOTIVE
RESEARCH AND DEVELOPMENT COMMAND
Warren, Michigan 48090

Approved for public release; distribution unlimited.
APPENDIX C (FORTRAN LISTINGS)

TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1</td>
<td>INTRODUCTION</td>
<td>C-3</td>
</tr>
<tr>
<td>C.2</td>
<td>MAIN PROGRAM, Labeled COMMONS AND BLOCK DATA</td>
<td>C-4</td>
</tr>
<tr>
<td></td>
<td>Main Program</td>
<td>C-5</td>
</tr>
<tr>
<td></td>
<td>Labeled Commons</td>
<td>C-7</td>
</tr>
<tr>
<td></td>
<td>Block Data</td>
<td>C-10</td>
</tr>
<tr>
<td>C.3</td>
<td>INITIALIZATION AND INPUT PROCESSING ROUTINES</td>
<td>C-17</td>
</tr>
<tr>
<td></td>
<td>CSCORE, Subroutine</td>
<td>C-18</td>
</tr>
<tr>
<td></td>
<td>INALT, Subroutine</td>
<td>C-21</td>
</tr>
<tr>
<td></td>
<td>INATTR, Subroutine</td>
<td>C-27</td>
</tr>
<tr>
<td></td>
<td>INATR2, Subroutine</td>
<td>C-33</td>
</tr>
<tr>
<td></td>
<td>INFUNC, Subroutine</td>
<td>C-38</td>
</tr>
<tr>
<td></td>
<td>INIT, Subroutine</td>
<td>C-42</td>
</tr>
<tr>
<td></td>
<td>INPROT, Subroutine</td>
<td>C-45</td>
</tr>
<tr>
<td></td>
<td>INSPEC, Subroutine</td>
<td>C-52</td>
</tr>
<tr>
<td></td>
<td>INVEHC, Subroutine</td>
<td>C-55</td>
</tr>
<tr>
<td>C.4</td>
<td>SOLUTION ROUTINES</td>
<td>C-65</td>
</tr>
<tr>
<td></td>
<td>COMPAT, Function</td>
<td>C-66</td>
</tr>
<tr>
<td></td>
<td>DIMENS, Subroutine</td>
<td>C-69</td>
</tr>
<tr>
<td></td>
<td>FPOWER, Subroutine</td>
<td>C-80</td>
</tr>
<tr>
<td></td>
<td>GENVEH, Subroutine</td>
<td>C-85</td>
</tr>
<tr>
<td></td>
<td>MOBILE, Subroutine</td>
<td>C-93</td>
</tr>
<tr>
<td></td>
<td>OKCMP, Function</td>
<td>C-96</td>
</tr>
<tr>
<td></td>
<td>PHIT, Function</td>
<td>C-100</td>
</tr>
<tr>
<td>C.5</td>
<td>OUTPUT ROUTINES</td>
<td>C-101</td>
</tr>
<tr>
<td></td>
<td>OUTARR, Subroutine</td>
<td>C-102</td>
</tr>
<tr>
<td></td>
<td>OUTCOM, Subroutine</td>
<td>C-105</td>
</tr>
<tr>
<td></td>
<td>OUTENG, Subroutine</td>
<td>C-108</td>
</tr>
<tr>
<td></td>
<td>OUTPER, Subroutine</td>
<td>C-112</td>
</tr>
<tr>
<td></td>
<td>OUTVEH, Subroutine</td>
<td>C-117</td>
</tr>
<tr>
<td></td>
<td>TABVAL, Subroutine</td>
<td>C-118</td>
</tr>
</tbody>
</table>
## TABLE OF CONTENTS (concluded)

### C.6. UTILITY ROUTINES

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARMULT, Subroutine</td>
<td>C-123</td>
</tr>
<tr>
<td>COPY, Subroutine</td>
<td>C-124</td>
</tr>
<tr>
<td>DVAL, Function</td>
<td>C-125</td>
</tr>
<tr>
<td>DVAL2, Function</td>
<td>C-126</td>
</tr>
<tr>
<td>ERR, Subroutine</td>
<td>C-127</td>
</tr>
<tr>
<td>FVAL1, Function</td>
<td>C-128</td>
</tr>
<tr>
<td>IATTR, Function</td>
<td>C-129</td>
</tr>
<tr>
<td>IDVAL, Function</td>
<td>C-130</td>
</tr>
<tr>
<td>IDVAL2, Function</td>
<td>C-131</td>
</tr>
<tr>
<td>INITVL, Subroutine</td>
<td>C-132</td>
</tr>
<tr>
<td>INLIST, Function</td>
<td>C-133</td>
</tr>
<tr>
<td>IOKVAL, Function</td>
<td>C-134</td>
</tr>
<tr>
<td>IRFIND, Function</td>
<td>C-135</td>
</tr>
<tr>
<td>ITABLE, Function</td>
<td>C-136</td>
</tr>
<tr>
<td>IVAL, Function</td>
<td>C-137</td>
</tr>
<tr>
<td>IZERO, Subroutine</td>
<td>C-138</td>
</tr>
<tr>
<td>LSERCH, Function</td>
<td>C-139</td>
</tr>
<tr>
<td>NEWREC, Function</td>
<td>C-140</td>
</tr>
<tr>
<td>NEWREL, Function</td>
<td>C-141</td>
</tr>
<tr>
<td>NEWVEH, Function</td>
<td>C-142</td>
</tr>
<tr>
<td>NLIST, Function</td>
<td>C-143</td>
</tr>
<tr>
<td>NULIFY, Subroutine</td>
<td>C-144</td>
</tr>
<tr>
<td>OKVAL, Function</td>
<td>C-145</td>
</tr>
<tr>
<td>QUEUE, Subroutine</td>
<td>C-146</td>
</tr>
<tr>
<td>STONAM, Subroutine</td>
<td>C-147</td>
</tr>
<tr>
<td>STORE, Subroutine</td>
<td>C-148</td>
</tr>
<tr>
<td>SUM, Function</td>
<td>C-149</td>
</tr>
<tr>
<td>SUMCMP, Function</td>
<td>C-150</td>
</tr>
<tr>
<td>SUMPD2, Function</td>
<td>C-151</td>
</tr>
<tr>
<td>SUMPRD, Function</td>
<td>C-152</td>
</tr>
<tr>
<td>TRANSF, Function</td>
<td>C-153</td>
</tr>
<tr>
<td>UNHOOK, Subroutine</td>
<td>C-154</td>
</tr>
<tr>
<td>ZERO, Subroutine</td>
<td>C-155</td>
</tr>
</tbody>
</table>

**Distribution List**

DD Form 1473
C.1: INTRODUCTION

This appendix contains the FORTRAN program listings for the COMPEND model. The listings are organized as follows:

- Section C-2 contains the main program, a listing of all labeled COMMON blocks (all other subprogram listings contain only the common block labels), and the BLOCKDATA subprogram.
- Section C-3 contains initialization and input processing routines.
- Section C-4 contains the main solution routines, including those used to estimate vehicle engineering and performance parameters.
- Section C-5 contains the output routines.
- Section C-6 contains utility routines.
C.2: MAIN PROGRAM, LABELED COMMONS AND BLOCK DATA

This section contains the following listings:

- The main program.
- A consolidated listing of all labeled COMMON blocks (subprogram listings in other sections of this report contain only the block labels for COMMON blocks used in that subprogram).
- The BLOCK DATA subprogram.
MAIN PROGRAM

COMBAT VEHICLE DEFINITION MODEL

This program produces a description of an armored combat vehicle of a specified class which meets the user's specifications. As part of his specifications the user may indicate various performance criteria that the vehicle must meet. He may also indicate various engineering parameters describing its components. Furthermore he may indicate which specific components are to be used in the concept vehicle. The program uses a description of what component types and how many of each are needed to produce a combat vehicle in the specified class. It also reads in a list of alternative components of each type it can use in assembling the concept vehicle. It then uses a generate-and-test procedure to assemble components which will produce a finished vehicle meeting the user's specifications.

For input and output the program uses the following data sets:

<table>
<thead>
<tr>
<th>DSRN</th>
<th>I/O</th>
<th>File Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I</td>
<td>DATA USED IN DESCRIBING FUNCTIONAL RELNS</td>
</tr>
<tr>
<td>2</td>
<td>I</td>
<td>EXISTING VEHICLE FILE</td>
</tr>
<tr>
<td>3</td>
<td>I</td>
<td>PROTOTYPE VEHICLE FILE</td>
</tr>
<tr>
<td>4</td>
<td>I</td>
<td>ALTERNATIVE COMPONENT FILE</td>
</tr>
<tr>
<td>5</td>
<td>I</td>
<td>USER SPECIFICATION FILE</td>
</tr>
<tr>
<td>6</td>
<td>O</td>
<td>ERROR MESSAGES</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>OUTPUT FILE</td>
</tr>
<tr>
<td>8</td>
<td>O</td>
<td>ECHO OF USER SPECS &amp; DATA HEADERS</td>
</tr>
</tbody>
</table>

1
LOGICAL ALTV
DATA ALTV/ FALSE./

2
PERFORM INITIALIZATIONS
CALL INIT

3
READ & STORE FUNCTIONAL RELATION DATA FILE
CALL IMPFNC
MAIN PROGRAM

C READ & STORE PROTOTYPE VEHICLE DEFINITION FILE
  CALL INPRT
C READ & STORE ALTERNATIVE COMPONENT FILE
  CALL INALT
C READ & STORE USER SPECIFICATIONS IMAGING EXTERNAL FORM
  CALL INSPEC
C EXTRACT ANY INFORMATION NEEDED FROM EXISTING VEHICLE FILE TO
C PRODUCE FINAL INTERNAL DESCRIPTION OF USER SPECIFICATIONS
  CALL INVEHC
C GENERATE "BEST" CONCEPT VEHICLE USING A BACKTRACK PROGRAMMING
C APPROACH
  CALL GENVEH
C OUTPUT A DESCRIPTION OF THIS CONCEPT VEHICLE
  CALL OUTVEH
C
  STOP
C
  END
COMMON BLOCKS

COMMON /VEH/ VEHICLE(72,3), HULL(48,9), TURRET(42,9), MAINGN(44,9),
  + MACGUN(40,10), MISGUN(40,7), ADGUN(40,6), RANGER(28,7),
  + SENSQR(32,9), STABLE(30,6), GUNCTR(28,9), AMMO(30,30),
  + AMMOC(26,9), ENGINE(40,8), TRANSM(32,8), FINLDR(28,11),
  + ROADWH(36,10), SPRING(30,10), TRACK(34,8), SKIRT(28,8),
  + FUEL(28,10), FUELC(26,9), CREW(26,5), CARGO(26,5), ELECTR(26,6),
  + COMM(28,9), FIREX(28,10), ENVIRC(26,9), DIAGNS(26,5),
  + SIGSUP(26,26), SMOKEG(26,10), EWSYS(26,10), RELN(3,50),
  + IPSPECV, ICANDV, IALT

DIMENSION IVECLE(72,3), IHULL(48,9), ITURRET(42,9), FMAING(44,9),
  + FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), ITRANGR(28,7),
  + ISENSR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),
  + IAMMOC(26,9), IENGINE(40,8), ITRANSM(32,8), IFINDR(28,10),
  + IROADWH(36,10), ISPGRN(30,10), ITRACK(34,8), ISKIRTR(28,8),
  + IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),
  + IFIREX(26,9), IFIREX(28,10), IFIREX(26,9), IDFIREX(26,9),
  + IDIAGNS(26,5), ISIGSUP(26,26), ISMOKEG(26,10), IEWSYS(26,10),
  + IRELN(3,50), VFILE(1), IVFILE(1)

EQUIVALENCE (IVECLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
  + ITURRET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)),
  + (MISGUN(1), FMISGUN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
  + TRANGR(1)), (SENSQR(1), ISENSQR(1)), (STABLE(1), ISTABLE(1),
  + (GUNCTR(1), IGNCTR(1)), (AMMOC(1), IAMMO(1)), (AMMOC(1),
  + IAMMOC(1), IENGINE(1), IENGINE(1)), (TRANSM(1), ITRANSM(1)),
  + IFINDR(1), IFINDR(1), IFINDR(1), ITRACK(1), ISKIRTR(1),
  + ISPGRN(1), ISPGRN(1), ISPGRN(1), ISPGRN(1), ISPGRN(1), ISPGRN(1),
  + (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
  + ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECTR(1),
  + (COMM(1), ICOMM(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),
  + IFIREX(1)), (DIAGNS(1), IDIAGNS(1)), (SIGSUP(1), ISIGSUP(1),
  + SMOKEG(1), ISMOKEG(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))

EQUIVALENCE (VECLE(1), VFILE(1), IVFILE(1))

COMMON/VPRT0/ FLDNAM(1049), RECNUM(33), DEFAULT(1049),
  + CMPCON(31), ICMPPTR(31), ICMPK(31), IFILP(33), IFLDPT(34),
  + IFLDPT(1049), IFLDV(1049), LSTOUT, LSTRES(33), MAXNUM(31),
  + MINNUM(31), NALTC(31), NFLDS(33), NRECS(33), OUTNAME(7,450),
  + MINNUM(31), NALTC(31), NFLDS(33), NRECS(33), OUTNAME(7,450),
COMMON BLOCKS

+ UNITS(1049), ICMATR, IEPARS, IPPARS, IOUTF(450)
REAL*8 FLONAM, RECNUM, DEFALT
COMMON / JPARS/
+ JACCEL, JACCUR, JACOST, JADGUN, JAI, JAMMO, JAMMED, JAMTP,
+ JANGLE, JAMTP, JBEATF, JBEATF, JB1, JBORE, JBRAKE, JCALIB,
+ JCARO, JCMPLX, JCOMO, JCONLR, JCOVER, JCREW, JCTLSY, JDOMP,
+ JOELD, JDIAGN, JDPRES, JDOTNG, JDWIND, JDWNP, JD1, JD2, JD3,
+ JECOST, JEFFIC, JEPWR, JEC, JERE, JEHOB, JENG, JENV,
+ JERPOT, JERAND, JEWSYS, JECAP, JFGLA, JFINDR, JFIREX, JFRATE,
+ JFUELC, JFUELC, JFUELC, JFUELC, JFUELC, JGAMO, JGAMU, JGHPTN, JHPPRS,
+ JGUI, JHMCVE, JHP, JHULL, JKI, JLOAD, JLOC, JMACG, JMACG,
+ JMA, JMAKUF, JMAT, JMATUR, JMINW, JMINST, JMODEL, JMUZL,
+ JMXPD, JNNAT, JNCRW, JNFWG, JNRDS, JNRDSH, JNRB, JNRVSG,
+ JNU, JOCOST, JOCIAH, JEPF, JERSN, JPHIT, JPHITS,
+ JPK, JPSLP, JPRANG, JPRANG, JRCST, JREL, JRCST, JRCST,
+ JRTIME, JSAIC, JSENSR, JSHPTN, JSIGSP, JSKHT, JSKT, JSTKTH,
+ JSPQ, JSMC, JSPD, JSPD, JSPRNG, JSTBSY, JSTEMP, JSTEM2
COMMON / JPARS1/
+ JSTIME, JUSPR, JST, JTBLEN, JTFIR, JTFIR, JTHFL, JTFJ,
+ JTONFT, JTRACK, JTRAD, JTRDE, JTRNFQ, JTB,
+ JTTF, JUTS, JUTU, JUAVEJ, JUVEJ, JVACOS, JVCM, JVHT,
+ JVLEN, JVENMR, JVENMR, JVENMR, JVENMR, JVENMR, JVENMR,
+ JVHT, JWEPK, JXEPK, JXHTRV, JXHT, JXH, JXTP, JX1, JX2,
+ JX3, JX4, JX5, JX6, JX7, JX8, JX9, JX10, JX21,
+ JY22, JY23, JY3, JY4, JY5, JY6, JY7, JY8, JY9, JY10,
+ JZ11, JZ12, JZ13, JZ14, JZ15, JZ16, JZ17, JZ18, JZ19,
+ JID, JNEXT, JTRD, JSCF, JSCF, JSCF, JSCF, JSCF,
+ JSIGMA, JSIGMA, JSIGMA, JSIGMA, JSIGMA
COMMON / NPARS/
+ NPAR, NPAR, NPAR, NPAR, NPAR, NPAR, NPAR, NPAR,
+ NPARAB, NPARB, NPARB, NPARB, NPARB,
+ NCMPT1, NCMPT1, NCMPT1, NCMPT1, NCMPT1,
+ NWTRC, NWTRC, NWTRC, NWTRC, NWTRC,
+ NRANGE, NRANGE, NRANGE, NRANGE, NRANGE,
+ NTKL, NTKL, NTKL, NTKL, NTKL,
+ NUTRL, NUTRL, NUTRL, NUTRL, NUTRL,
COMMON BLOCKS

COMMON /OUTPUT/ XNAT(32), XMANUF(32), XMODEL(32), NUMBER(32),
+ WEIGHT(32), PCWT(32), VOLUME(32), PCVOL(32), DWIN(32),
+ MATUR(32), ICmplx(32), RELAB(32), INDX(32), NPROD(32),
+ RTIME(32), KOUT(32), PMCGT, DF(8,31)
LOGICAL*1 DF
REAL*8 XNAT, XMANUF, XMODEL
COMMON /AUX/ BL8, CASHES, BLS, IRELTP(15), ITYPES(4), MULT, NULL,
+ PLUS, REName(15), SMINUS, TUNITS(3,50)
REAL*8 BLR, DASHES
EQUIVALENCE (FNULL,NULL), (IBLS,BLS)
COMMON /ARRAYS/ DETERG(3,2,9), NATION(160), MANUF(160),
+ MODEL(160), PHITM(3,2,3,3), PHITSI(3,2,3,3),
+ PK(3,2,3,3), PPENET(3,2,3,3), TRNRAD(4,3), SIGMA(2,3,16),
+ SIGMA2(2,3,16), APRLAB(5,75), IARDF(9,28), IARDF2(16,13)
REAL*8 MANUF, MODEL, NATION
DIMENSION ARRAY(1)
EQUIVALENCE (ARRAY(1), DETERG(1))
COMMON /JARS2/ JARST, JARLINK, JARUM, JARRPT,
+ JDNAME, JFLDRC, JLEVPT, JINDEX, JRECTP
COMMON /SP/ ATRENG(200), ATRPER(34), VEHCAT, VEHCM(31),
+ VEHENG(200), VFHPER(34), VEHT, VEHEVL(5), NUMCMP(31),
+ TYPcmp(31), INDcmp(31), LSTcmp, TYPENG(200), RELENG(200),
+ MULENG(200), VL1ENG(200), ADDENG(200), VL2ENG(200), INDENG(200),
+ LSTENG, RELPER(34), MULPER(34), VLPER(34), ADDPER(34),
+ VL2PER(34), STLPER
REAL*8 ATRENG, ATRPER, VEHCAT, VEHCM, VEHENG, VEHPER
COMMON /SP/ VEH(10), ICMP(31), ICOLS(15), IRECS(6,250),
+ JCOLUMN, JDONE, JNXT, JSPEC, JTYP, JVEH, LSTVEH, LVEH, NRECNI, NVEH, NVRF
REAL*8 VEH
COMMON /DATA/ ACCEL(4,5), CMPWTS(31), DENSITY(8),
+ MAXWHL, MINWHL, PHNULL, PNULL(4,3,2,4), PKDAT(3,3,2,3),
+ PTUP(4,3,2,4), RMPHTH(4), RLTMAX, RLTMIN,
+ RSHPTN(4), RVEHT(5), RTKLEN(4), RWTRTV(4),
+ SIGMV(2,3,3,3), SIGMV(3,2,3,2),
+ SLOP(4,5), SPEED(4,2), VRISE(4,5,4,3,2), WIDMAX, XMPG(4,5,4)
COMMON /DATA/ RDISPR(3)
BLOCK DATA

COMMON /VEH/ VEHICLE(72,3), HULL(48,9), TURRET(42,9), MAINLINE(44,9),
+ MACGIN(40,10), MISGIN(40,7), ADGUN(40,6), RANGER(28,7),
+ SENSOR(32,9), STABLE(37,6), UNCTIL(28,9), AMMO(30,30),
+ AMMOC(26,9), ENGINE(40,8), TRANSM(32,8), FINLDR(28,10),
+ ROADWH(36,10), SPRING(30,10), TRACK(34,8), SKIRT(28,8),
+ FUEL(28,11), FUEL(26,9), CREW(26,5), CARGO(26,5), ELECTR(26,5),
+ COMMOD(28,9), FIREX(28,10), ENVIRO(26,9), DIAGNS(26,5),
+ SIGSU(26,24), SMOKE(26,10), EWSYS(26,10), RELN(3,5),
+ ISPECV, ICANDV, IALTV

DIMENSION [VEHICLE(72,3), IHULL(48,9), ITURRET(42,9), FMAINLINE(44,9),
+ FMACGIN(40,11), FMISGUN(40,7), IADGUN(40,6), IRANGR(23,7),
+ ISNS(32,11), ISTATE(30,6), IGROUP(28,9), IMAMMO(30,30),
+ IAMMOC(26,9), IEINIG(40,8), ITMINS(32,8), IFINDR(28,10),
+ IPBDWH(36,10), ISPEN(30,10), ITRACK(34,8), IISKIR(23,8),
+ IFUEL(28,10), IFUEL(26,9), ICREW(26,5), ICARGO(26,5),
+ IELECTR(26,6), ICOMMOD(28,9), IFIREX(28,10), IEVIR(26,9),
+ IDIAGNS(26,5), ISIGSU(26,24), ISMOKE(26,10), IEWSYS(26,10),
+ IRELN(5,5), IFIELD(1), IFIELD(1)

EQUIVALENCE (VEHICLE(1), IVEHICLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
+ ITURRET(1)), (MAINLINE(1), FMAINLINE(1)), (MACGIN(1), FMACGIN(1),
+ (MISGUN(1), FMISGUN(1), (ADGUN(1), IADGUN(1)), (RANGER(1),
+ (IRANGER(1)), (SENSOR(1), ISNS(1)), (STABLE(1), ISTATE(1)),
+ (GROUP(1), IGROUP(1), (AMMOM(1), IAMMO(1)), (AMMOC(1),
+ (IAMMOC(1), IENGINE(1), IENGINE(1)), (TRANSM(1), ITRANS(1)),
+ (FINLDR(1), IFINDR(1)), (ROADWH(1), IPBDWH(1)), (SPRING(1),
+ ISPEN(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), IISKIR(1)),
+ (FUEL(1), IFUEL(1)), (FUEL(1), IFUEL(1)), (CREW(1),
+ ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECTR(1),
+ (COMMOD(1), ICOMMOD(1)), (FIREX(1), IFLREX(1)), (ENVIRO(1),
+ IEVIR(1)), (DIAGNS(1), IDIAGNS(1)), (SIGSU(1), ISIGSU(1)),
+ (SMOKE(1), ISMOKE(1)), (EWTSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))

EQUIVALENCE (VEHICLE(1), IVEHICLE(1), IFIELD(1), IFIELD(1))

COMMON/VRM/ /FLDNAM(1,49), /RFICNAM(33), /DEFAULT(10,49),
+ CMPCOD(31), ICMPT(31), ICMPRT(31), IFILPT(33), IFILPT(34),
BLOCK DATA

+ IFLDTP(1349), IFLDVL(1049), LSTOUT, LSTREC(33), MAXNUM(31),
+ MINNUM(31), NALTCS(31), NFLO5(33), NRECS(33), OUTNAM(7,450),
+ UNITS(1049), ICMATR, IEPARS, IPPARS, IOUTF(450)

REAL*8 FLDNAM, RECNAM, DFAULT

COMMON /JPARS/
+ JACCEL, JACCUR, JACOST, JADGUN, JAI, JAMMO, JAMMOC, JAMOTP,
+ JANGLE, JAMTP, JBEATF, JBEATR, JBI, JBORE, JBAKE, JCALIB,
+ JCARGO, JCFLXJ, JCCMOC, JCSTL, JCOVER, JCREW, JCTLSY, JDOMP,
+ JDELN, JDIAKN, JDPRES, JDTREN, JDWIN, JD1, JD2, JD3,
+ JECOST, JEFFIC, JFPPWR, JELFC, JELFCN, JEMOB, JENGIN, JENVR,
+ JEPRT, JERAMO, JERWSYS, JFCAP, JFGRJ, JFRD, JFRATE,
+ JFUEL, JFUELC, JFUELE, JFUELN, JGAMO, JGAMU, JGHPTR, JGPPRS,
+ JGUIDE, JHMCVE, JHPL, JHULL, JKI, JLOAD, JLOC, JMAC6G, JMACG,
+ JMAING, JMANUF, JMACT, JMATUR, JMINPT, JMISL, JMODEL, JMUSLE,
+ JMXPD, JNAT, JNCREW, JNFDW, JNRDWH, JNRTRN, JNRSVG,
+ JNUM, JOCOST, JOCIAH, JOHT, JPERF, JPERSON, JPHITM, JPHITS,
+ JPK, JPSLOP, JRAP, JRAPG, JRCOST, JRELBE, JRGANG, JROAD,
+ JRTIME, JSCAJ, JSENSE, JSHPTN, JSIGSP, JSKHT, JSKIR, JSKTHK,
+ JSLOPE, JSMOEK, JSPD1, JSPD2, JSPRG, JSTBSY, JSTEM, JSTEM2

COMMON /JPARS1/
+ JSTIME, JSUSP, JT, JTBLEN, JTFIR, JTFIR1, JTHFL, JTHFU,
+ JTONF, JTACK, JTRAD, JTRANS, JTRANS, JTRNQ, JTTB,
+ JUTF, JUTS, JUTU, JUURE, JUTE, JUTYPE, JUTYPE, JUTYPE, JVACOS, JVCMPX, JVH,
+ JVLEN, JVMAT, JVMMS, JVOCOS, JVOL, JVRPR, JVRPR, JVRPR,
+ JVWT, JWDENP, JWDENP, JWHTRV, JXH, JXTJ, JX1, JX2,
+ JX3, JX4, JX5, JX6, JX7, JX8, JX9, JY2, JY20, JY21,
+ JY22, JY23, JY3, JY4, JY5, JY6, JY7, JY8, JY9, JZH, JZI, JZ10,
+ JZ11, JZ12, JZ13, JZ14, JZ15, JZ2, JZ3, JZ4, JZ5, JZ7, JZ8, JZ9,
+ JD, JNEN, JNENP, JSCORE, JVALUE, JKEY, JLT, JPENTR,
+ JSIGMA, JSIGM2, J30SPD

COMMON /NPARS/
+ NAMC, NAMC2, NAMTP, NADMS, NAMTH, NARRI,
+ NARLAB, NARRS, NASPEC, NCOMAT, NCSPCS, NCMPPT,
+ NCMP1, NCMPS, NCMP2, NDAYNT, NDETTP, NDISPR, NESPCS, NEVALC,
+ NFDPC, NEGAE, NGRAF,
+ NWHTRV, NRDWHL, NORDT2, NORDTP, NOUT, NOUTD, NPSPCS,
BLOCK DATA

+ N_RANGE, N_RECTP, NRECTPI, NRELOP, NSHPTN, NSTABL, NSMTGT, VTERTP,
+ NTKLEN, NENGTP, NTYPES, NUTRNI, NUTRNS, NVEHTW, NXY
COMMON /OUTPUT/ XNAT(32), XMANUF(32), XMODEL(32), NUMBER(32),
+ WEIGHT(32), PCWT(32), VOLUME(32), PCVOL(32), DNUP(32),
+ MATUP(32), ICPLX(32), RELAB(32), INDEX(32), NPROD(32),
+ RDTIME(32), KOUT(32), FMCGT, DF(8,31)
LOGICAL*1 DF
REAL*8 XNAT, XMANUF, XMODEL
COMMON /AUX/ BL8, CASHES, BLS, IRELTP(15), ITYPES(4), MULT, NULL,
+ PLUS, RELNAME(15), SMINUS, TUNITS(3,5)
REAL*8 BL8, CASHES
EQUIVALENCE (FNUL, NULL), (IBLS, BLS)
COMMON /ARAYS/ DETERG(3,2,9), NATION(160), MANUF(160),
+ MODEL(160), PHITM(3,2,3,3), PHITS(3,2,3,3),
+ PK(3,2,3,3), PPNEN(3,2,3,3), TRNRA(4,3), SIGMA(2,3,3,16),
+ SIGMA2(2,3,3,16), APRLAB(5,75), IARDEF(9,28), IARDEF2(6,13)
REAL*8 MANUF, MODEL, NATION
DIMENSION ARRAY(1)
EQUIVALENCE (APRAY, DETERG(1))
COMMON /JPARS2/ JARSIZ, JARLST, JARNUM, JARRPT,
+ JDNAM, JFLCRC, JLEVT, JNELVL, JRECTP
COMMON /SPECS2/ VEH(10), ICMP(31), ICOLS(10), IREC(6,250),
+ JCOL, JDONE, JNXT, JSPEC, JTYPE, JVEH, LSTVEH, LVEH, NIREC, NVEHS, NVR
REAL*8 VEH
COMMON /DATA/ ACCEL(4,5), CMPWTS(31), DENSITY(8),
+ MAXWHL, MINWHL, PPHULL(4,3,2,2,4), PKDAT(3,3,2,3),
+ PTURRP(4,3,2,2,4), RARMTH(4), RLMAX, RLMIN,
+ PSPTN(4), PVHWT(5), RTKLEN(4), RWHTRV(4),
+ SIGMF(2,3,3,3), SIGMV(2,3,3,2),
+ SLOPE(4,5), SPEED(4,2), VRIDE(4,5,4,3,2), WIDMAX, XMPS(4,5,4)
COMMON /DATA2/ RDISP(3)
C
DIMENSION KARDEF(9,14)
EQUIVALENCE (KARDEF(1,1), IARDEF(1,14))
C
DATA
BLOCK DATA

+ JACCEL/52/, JACCUR/26/, JACOST/4/, JADGUN/6/, JAI/25/,
+ JAMMO/11/, JAMMOC/12/, JAMTOP/29/, JANGLE/27/, JARMTP/28/,
+ JBETA/33/, JBEATR/34/, JBI/25/, JBORE/29/, JBRAKE/56/,
+ JCALIB/27/, JCARGO/23/, JCMLPL/9/, JCMOMO/25/, JCOOLR/27/,
+ JCORDER/34/, JCREW/22/, JCTLSY/10/, JDAVE/27/, JDELY/38/,
+ JDPRES/28/, JDRNAG/28/, JDTNRG/29/, JDNWUP/45/, JDOCON/2/,
+ JD1/29/, JD2/29/, JDC3/27/, JECOST/22/, JEFFIC/26/,
+ JEPWR/18/, JELEC/24/, JELTEN/36/, JEMOB/19/, JENGIN/13/
+ JENVR/27/, JEPRTC/20/, JERMAD/21/, JEWSYS/31/, JFACAP/26/,
+ JFGAL/26/, JFIND/15/, JFIREX/26/, JFRATE/33/, JFUEL/20/,
+ JFUEL/21/, JFUELE/29/, JFUEL/28/, JGAME/37/, JGAMJ/36/,
+ JGHPTE/41/, JGPRES/43/, JGUIDE/28/, JHMOVE/35/, JHP/26/,
+ JHULL/1/, JK/31/, JLOAD/30/, JLOC/17/, JMACG/4/,
+ JMAJ/28/, JMAG/3/, JMANIF/14/, JMATER/26/, JMATUR/8/,
+ JMINWT/38/, JMISS/5/, VMODEL/15/, JMOUDL/25/, JMSPO/48/,
+ JMSW/13/, JSCREW/25/, JNFWOD/27/, JNRD/26/, JNRWON/25/,
+ JNRTN/26/, JNRVSG/28/, JNUN/12/, JOCOST/5/, JODIAN/42/,
+ JOUTH/44/, JPERF/26/, JPERSN/26/, JPHITM/39/, JPHITS/38/,
+ JPK/40/, JPSLCP/57/, JRENG/50/, JRENGE/51/, JRCAP/6/

DATA

+ JREL/10/, JRNAD/16/, JRTIME/7/, JSADJ/27/,
+ JSEN/8/, JSHP/42/, JSIGSP/29/, JSSH/26/, JSKRT/19/,
+ JST/27/, JSTP/53/, JSMOK/30/, JSP/58/, JSPD2/59/,
+ JSPRGT/28/, JSTESY/9/, JSTEM2/33/, JSTIME/31/,
+ JSTUS/17/, JTW/29/, JTRLEN/26/, JTF/32/,
+ JTFIR/31/, JTHEL/32/, JTHF/31/, JTON/69/, JTRACK/18/,
+ JTRAD/55/, JTPANS/14/, JTRATE/54/, JTRNRQ/30/,
+ JTVB/39/, JTV/26/, JTVS/37/, JTU/38/, JTURET/2/,
+ JTYPE/25/, JTYPEE/26/, JUCO/65/, JUCMPL/62/, JVHT/35/,
+ JVOEN/37/, JWXMATR/61/, JXMBF/63/, JXCO/66/, JXVOL/37/,
+ JXVPHR/64/, JXVQ/34/, JXVWD/36/, JXWT/33/, JXWDPN/45/,
+ JXWDEP/47/, JXWHTV/25/, JXT/2/, JXH/27/,
+ JXTP/27/, JXV/39/, JXV/30/, JX3/29/, JX4/27/, JX5/35/,
+ JXG/38/, JX/29/, JX/27/, JY/3/6/, JYGC/68/, JYH/26/,
+ JY2/39/, JY2/40/, JY2/41/, JY2/34/, JY2/35/, JY3/40/,
BLOCK DATA

+ JZH/25/, JZ1/32/, JZ10/36/, JZ11/37/, JZ12/34/, JZ13/33/,
+ JZ14/31/, JZ15/33/, JZ2/35/, JZ4/33/, JZ5/32/, JZ7/49/,
+ J28/43/, JZ9/44/, JZ3/45/
DATA
+ JKY/32/, JLT/67/, JENETR/60/, JSIGMA/34/, JSIGM2/27/,
+ J3OSPD/49/,
+ JID/24/, JNEXT/1/, JRELCP/2/, JVALUE/3/, JRECTP/1/, JFJ DRC/2/,
+ JNLEVL/3/, JLEVPT/4/, JONAME/5/, JARSIZ/3/, JARLST/4/, JARRPT/5/,
+ JARNUM/6/, JSCORE/23/, JDONE/6/.
DATA
+ NARMCI/16/, NARMCI2/16/, NARMTP/8/, NADIMS/28/, NARMTH/4/,
+ NARLAB/75/, NARRS/13/,
+ NASPEC/2/, ACCMAT/25/, NCSPCS/31/, NCMPTP/31/, NCMPT1/32/,
+ NCMPS/160/, NCMPTL/12/, NDAYNT/2/, NDETP/3/, NDISPR/3/,
+ NEPCS/200/, NEVALC/5/, NEFDPC/994/, NGEAR/4/, NGP ADE/ 2/,
+ NIREC/6/, NWHTRV/4/, NRDWHL/3/, NORT2/2/, NORTD/3/,
+ NOUT/450/, NOUTND/7/, NPSPCS/34/, N RANGE/3/, NRECTP/33/,
+ NRNP/34/, NRELOP/15/, NNSTPN/4/, NSTABL/3/, NSTMGT/2/,
+ NTERPT/2/, NTKLEN/4/, NENGTP/4/, NTYPE4/4/, NUTRNI/3/,
+ NUTRMS/50/, NVEWT/5/, NYX/2/
DATA RDI SPR/100C. , 2000., 3000./
DATA NRECS/3, 9, 9, 9, 10, 7, 6, 7, 9, 6, 9, 9, 9, 8, 8, 10, 10, 10, 8, 8,
+ 10, 9, 5, 6, 9, 9, 10, 5, 24, 10, 10, 50/
DATA NFLOD/72, 48, 42, 44, 40, 40, 40, 28, 32, 30, 28, 33, 26, 49, 32,
+ 28, 36, 30, 34, 28, 28, 26, 26, 26, 28, 28, 26, 26, 26, 26, 26, 3/.
DATA ISPFCV/1/, ICANDV/2/, IALT/3/.
DATA RLS/" /, RLS/" /, RMTL/" **/ , NULL/-1/ , PLUS/" +/ , SMNUM/" -/- /, + DASHES/"----------/,
DATA RELNAM/" LT/ , LE/ , EQ, " NE/ , GE/ , GT/ , " <\', '<\', '<\', '<\', '<\', '<\',
+ '>' , '>' , '>' , '>' , '>' , '>' , "/
DATA IYPES/" R/ , I/ , RA/ , C8/,

C-14
C.3: INITIALIZATION AND INPUT PROCESSING ROUTINES

This section contains listings of initialization and input processing routines. See section C-2 for listings of labeled COMMON blocks referred to in these listings.
SUBROUTINE CSORE

ROUTINE TO COMPUTE THE SCORE OF THE VARIOUS ALTERNATIVE COMPONENTS AND TO ORDER THE COMPONENTS OF EACH TYPE IN DECREASING ORDER BY SCORE.

COMMON /VEH/
DIMENSION IVECLE(72,3), IHULL(48,9), ITURRET(42,9), FMAING(44,9), 
+ FMAGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7), 
+ ISENSR(32,9), ISTABL(3,6), IGNCTL(28,9), IAMMO(30,30), 
+ IAMMO(26,9), IENGIN(40,8), ITBUS(32,8), IFINDR(28,10), 
+ IROADV(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8), 
+ IFUEL(28,10), IFUEL(26,9), ICREW(26,5), ICARGO(26,5), 
+ IFLEX(26,6), ICCOMMO(28,9), IFIREX(28,10), IEENVIR(26,9), 
+ IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,13), IEWSYS(26,11), 
+ IRELN(3,50), VFILE(11), IVFILE(1)

EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1), 
+ ITURRET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)), 
+ (MISGN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1), 
+ IRANGR(1)), (SENSR(1), ISENSR(1)), (STABLE(1), ISTABL(1)), 
+ (GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1), 
+ IAMMOC(1)), (ENGIN(1), IENGIN(1)), (TRANS(1), ITRANS(1), 
+ (FINLDR(1), IFINDR(1)), (ROADW(1), IROADV(1)), (SPRING(1), 
+ ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)), 
+ (FUEL(1), IFUEL(1)), (FUEL(1), IFUEL(1)), (CREW(1), 
+ ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)), 
+ (COMMO(1), ICCOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIR(1), 
+ IEENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSP(1), ISIGSP(1)), 
+ (SMOKE(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))

EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))

COMMON /VPROTO/
REAL *8 FLDNAM, RECNAM, DEFAULT
COMMON /JPARS/
COMMON /JPARS1/
COMMON /NPARS/
COMMON /AUX/
INPUT ROUTINES - CSCORE SUBROUTINE

ISN
12    REAL*8 BLB,DASHES
13    EQUIVALENCE (FNULL,NUL), (IBLS,BLS)
14    COMMON /SPECS/
15    REAL*8 ATRENG,ATRPER,VEHCAT,VEHCM,VEHENG,VEHPER
C
16    DIMENSION IRS(50),NEXT(50),SCORES(50)
C
17    ITE 50 ICMP=1,NCMPTP
C
18    GET ATTRIBUTES OF COMPONENT
19    NF =NFLDS(ICMP+1)
20    IOFF =IFILPT(ICMP+1)
C
21    ITE 50 COMPONENTS OF THAT TYPE
22    IREC = ICMPPT(ICMP)
23    LIST = NULL
24    J = 0
25    CALL NULIFY(NEXT,50)
26    CALL ZERO(SCORES,50)
27    10 IF (IREC EQ. NULL) GO TO 30
28    IOFFR = IOFF + NF*(IREC-1) - 1
29    SCORE = SUMPRC(VFILE(IOFFR+JEFPWR),VEHEVL,5)
30    VFILE(IOFFR+JSCORE) = SCORE
C
31    INSERT NEW COMPONENT IN LIST IN DECREASING ORDER
32    J = J + 1
33    IRS(J) = IREC
34    SCORES(J) = SCORE
35    IF (J .NE. 1) GO TO 14
36    NEXT(J) = LIST
37    12    LIST = J
38    20   GO TO 20
39    14 IF (SCORE .GT. SCORES(LIST)) GO TO 12
40    K = LIST
41    KP = K
INPUT ROUTINES - CSCORE SUBROUTINE

39 K = NEXT(K)
40 IF (K .NE. NULL .AND. SCORE .LE. SCORES(K)) GO TO 16
41 NEXT(J) = K
42 NEXT(KP) = J
43 IREC = IVFILE(IOFRR+JNEXT)
44 GO TO 10

C STRING TOGETHER COMPONENTS IN DECREASING ORDER BY THEIR SCORES
31 K = LIST
42 ICMPPT(ICMP) = LIST
47 IF (K .EQ. NULL) GO TO 50
49 IOFFR = IOFF + NF*(K - 1) - 1
50 IVFILE(IOFFR+JSCORE) = NEXT(K)
51 GO TO 32

C
52 CONTINUE
53 RETURN
54 END
INPUT ROUTINES – INALT C SUBROUTINE

SUBROUTINE INALT C

ROUTINE FOR READING AND STORING ALTERNATIVE COMPONENTS.

COMMON /VEH/
DIMENSION IVECLE(72,3),I HULL(48,9),ITURE T(42,9),FMA ING(44,9),
+ FMAGN(40,10),FMISGN(40,7),IAD GUN(40,6),IRANGE(28,7),
+ ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMOC(30,30),
+ IAMMOC(26,9),IGEN IN(4),8),ITRAN S(32,8),IFIN D(28,10),
+ IROADW(36,10),IPNRG(30,10),ITRACK(34,8),ISKIR T(28,8),
+ IFUEL(28,10),IFUEL C(26,9),ICREW(26,5),ICARGO(26,5),
+ IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
+ IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IENWSYS(26,10),
+ IRELN(3,50),VFLE(1),IVFLE(1)
EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),I HULL(1)),(TURRET(1),
+ ITURRE T(1)),(MAINGN(1),FMA IN(1)),(MACGun(1),FMACGN(1)),
+ (MAJGN(1),FMISGN(1)),(ADGun(1),IADGun(1)),(RANGE(1),
+ IRANGER(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
+ (GU NCTL(1),IGNCTL(1)),(AMMO(1),IAMMOC(1)),(AMMOC(1),
+ IAMMOC(1)),(ENGINE(1),IGEN IN(1)),(TRANSM(1),ITRAN S(1)),
+ (FIR NDR(1),IFIN D(1)),(ROADWH(1),IROADW(1)),(SP RING(1),
+ ISP NRG(1)),(TRACK(1),ITRACK(1)),(ISKIRT(1),ISKIR T(1)),
+ (FUEL(1),IFUEL C(1)),(FUEL C(1),IFUEL C(1)),(CRE W(1),
+ ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
+ (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
+ IENVYR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1),
+ (SMOKE(1),ISMOCKE(1)),(ENWSYS(1),IENWSYS(1)),(RELN(1),IRELV(1))
EQUIVALENCE (VEHICLE(1),VFLE(1),IVFLE(1))

COMMON /VPRTOP/,
REAL*8 FLDA, RCNAM, DFAULT
COMMON /JPARS/
COMMON /JPARS1/
COMMON /NPARS/
COMMON /AUX/
REAL*8 BLB, DASHES
INPUT ROUTINES - INALTC SUBROUTINE

ISN
13 EQUVALENCE (FNULL, NULL), (IBLS, BLS)
14 COMMON /ARRAYS/
15 REAL*8 MANIF, MODEL, NATION
16 DIMENSION ARRAY(1)
17 EQUVALENCE (ARRAY(1), DETRNG(1))
18 COMMON /JPARS2/
19 COMMON /SPECS/
20 REAL*8 ATRENG, ATRPER, VEHCAT, VEHCMPI, VEHENG, VEHPER

REAL*8 V, CMP, CMP2, ATTR
22 DIMENSION INT(12), IRECS2(12), VALS(12), IVALS(12), CVALS(2, 12)

C READ VEHICLE CATEGORY RECORD
23 READ (4, 100) V
24 100 FORMAT (A8)
25 IF (V NE. VEHCAT) CALL ERR(4,'INALTC', V, VEHCAT, 0)
26 CMP2 = BL8

C READ COMPONENT HEADER
27 READ (4, 105, END=90) CMP,(INT(I), I=1, NCMPLZ)
28 105 FORMAT (A8, 2X, I2, 8X, 12)
C CHECK WHETHER HAVE ENCOUNTERED A NEW TYPE OF COMPONENT
29 IF (CMP .EQ. CMP2) GO TO 5
30 N = 0
31 CMP2 = CMP

C DETERMINE HOW MANY NON-ZERO ALTERNATIVES THERE ARE
32 DO 10 I=1, NCMPLZ
33 IF (INT(I) .EQ. 0) GO TO 15
34 IF (INT(I) .NE. N+I) CALL ERR(5,'INALTC', CMP, N+I, INT(I))
35 CONTINUE
36 N = N + NCMPLZ
37 NAM = NCMPLZ
38 GO TO 20

C
39 15 I = I - 1
**INPUT ROUTINES — INALTC SUBROUTINE**

40  \( N = N + 1 \)
41  \( NA = 1 \)

C IDENTIFY RECORD AND COMPONENT TYPES; STORE NUMBER OF ALTERNATIVES
C OF THIS TYPE COMPONENT
42  DO 25 IRECTP=1,NRECTP
43  IF (CMP .EQ. RECNAME(IRECTP)) GO TO 27
44  25 CONTINUE
45  CALL ERR(9,'INALTC',CMP,0,0)
46  ICMP = IRECTP - 1
47  IF (ICMP .GE. 1) NALTCS(ICMP) = N

C DETERMINE OFFSET IN FILE IN WHICH RECORDS STORED
48  IOFF = IFILPT(IRECTP)

C DETERMINE NUMBER OF FIELDS PER RECORD IN THIS FILE
49  NF = NFLDS(IRECTP)

C GET NA NEW COMPONENT RECORDS AND QUEUE THEM ONTO THE CURRENT LIST
50  DO 30 IREC=1,NA
51      IR = NEWREC(IRECTP)
52      CALL STORE(INT(IREC),VFilen(irez),NF,JID,IR)
53      IRECS2(IREC) = IR
54  CALL QUEUE(IR,ICMPPT(IRECTP-1),IRECTP)
55  30 CONTINUE

C READ COMPONENT ATTRIBUTES
56  40 READ (4,110,END=92) ATTR
57  110 FORMAT (2X,AB)

C IS THIS THE END OF THIS SET OF ATTRIBUTES ?
58  IF (ATTR .EQ. BL8 .OR. ATTR .EQ. DASHES) GO TO 2
C FIND FIELD INDEX AND RECORD TYPE
59  IFLD = IATTR(ATR,CMP,IRECTP)

C IF ATTRIBUTE NAME IS UNRECOGNIZED, HAVE AN ERROR
60  IF (IFLD .NE. NULL) GO TO 45
61  CALL ERR(6,'INALTC',CMP,ATTR,0)
62  GO TO 40
INPUT ROUTINES - INALTC SUBROUTINE

C C DETERMINE TYPE OF ATTRIBUTE
63 45 KFLD = IFLOPT(IRECTP) + IFLD
64 ITYP = IFLDTPE(KFLD)
C C BACKSPACE RECORD AND READ VALUES USING FORMAT APPROPRIATE TO TYPE
65 BACKSPACE 4
66 DO 50 I=1,NTYPES
67 IF (ITYP .EQ. ITYPES(I)) GO TO 55
68 50 CONTINUE
69 CALL ERR(10,'INALTC*,CMP,ATTR,ITYP)
70 GO TO 2
C C BRANCH APPROPRIATE TO TYPE
71 55 GO TO (60,65,70,80),I
C C C READ & STORE REAL VALUES
72 60 READ (4,115) (VALS(J),J=1,NA)
73 115 FORMAT (12X,12(F8.0,2X))
74 BACKSPACE 4
75 READ (4,125) (CVALS(1,J),CVALS(2,J),J=1,NA)
76 DO 61 J = 1,NA
77 IF (CVALS(1,J) .EQ. BLS .AND. CVALS(2,J) .EQ. BLS) + VALS(J) = FNULL
78 61 CONTINUE
79 DO 62 J=1,NA
80 CALL STORE(VALS(J),VFIL,IOFF,NF,IFLD,IRECS2(J))
81 62 CONTINUE
82 GO TO 40
C C C READ AND STORE INTEGER VALUES
83 65 READ (4,120) (IVALS(J),J=1,NA)
84 120 FORMAT (12X,12(I8,2X))
85 BACKSPACE 4
86 READ (4,125) (CVALS(1,J),CVALS(2,J),J=1,NA)
87 DO 67 J = 1,NA
88 IF (CVALS(1,J) .EQ. BLS .AND. CVALS(2,J) .EQ. BLS)
INPUT ROUTINES - INALT C SUBROUTINE

15 SN
+ IVALS(J) = NULL
89  67 CONTINUE
90  68 DO J=1,NA
91  69 CALL STORE(IVALS(J),VFILE(IOFF),NF,IFLD,IRECS2(J))
92  68 CONTINUE
93  69 GO TO 40
C
C READ & STORE AN ARRAY OF REAL VALUES
C
C DETERMINE WHICH ARRAY FILE INFORMATION WILL BE STORED IN
94  70 DO IARR=1,NARRS
95  71 IF (IARDF2(JRECTP,IARR) .EQ. IRECTP .AND. IARDF2(JFLDRC,IARR)
96  72   + .EQ. IFLD) GO TO 73
97  71 CALL ERR(16,'INALT C',IRECTP,IFLD,ITYP)
C
C NUMBER OF ELEMENTS IN ARRAY
98  72 NV = IARDF2(JARSZ,IARR)
C OFFSET IN FILE CONTAINING ARRAYS
99   IOFFAR = IARDF2(JARRPT,IARR)
C GET NA VACANT ARRAYS OF TYPE IARR & STORE POINTERS TO THEM FROM THE
C CURRENT RECORD
100  75 DO J=1,NA
101  76 K = IARDF2(JARLSZ,IARR) + 1
102  77 IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INALT C',
103     + ATTR,IARDF2(JARNUM,IARR),IARR)
104   IARDF2(JARLSZ,IARR) = K
105  78 CALL STORE(K,VFILE(IOFF),NF,IFLD,IRECS2(J))
106  79 IRECS2(J) = K
107  80 CONTINUE
C READ NV VALUES FOR EACH CF THE NA ARRAYS AND STORE THEM
108  81 DO I=1,NV
109  82 READ (4,115) (IVALS(J),J=1,NA)
110  83 BACKSPACE 4
111  84 READ (4,125) (CVALS(1,J),CVALS(2,J),J=1,NA)
112  85 DO J = 1,NA
INPUT ROUTINES - INALT C SUBROUTINE

112 IF (CVALS(1,J) .EQ. BLS .AND. CVALS(2,J) .EQ. BLS)
+ VALS(J) = FNULL
113 CONTINUE
114 DO 77 J=1,NA
115 CALL STORE(CVALS(J),ARRAY(IOFFAR),NV,I,IRECS2(J))
116 77 CONTINUE
117 GO TO 40

C READ AND STORE CHARACTER INFORMATION (<= 8 CHARs)
118 READ (4,125) (CVALS(1,J),CVALS(2,J),J=1,NA)
119 FORMAT (12X,12(2A,2X))

C DETERMINE WHICH ARRAY FILE INFORMATION WILL BE STORED IN
120 DO 82 IARR=1,NARRS
121 IF (IARDF2(IJFLDRC,IARR) .EQ. IFLD) GO TO 83
122 CONTINUE
123 CALL ERR(16,'INALT C',IRECTP,IFLD,ITYP)

C OFFSET IN FILE CONTAINING CHARACTER STRINGS
124 IOFFCH = IARDF2(JARRPT,IARR)

C GET NA VACANT CHARACTER STRING RECORDS AND STORE POINTERS TO
C THEM FROM THE CURRENT RECORD. ALSO STORE CHARACTER STRINGS IN
C THE CHARACTER STRING RECORDS.
125 DO 85 J=1,NA
126 K = IARDF2(JARLST,IARR) + 1
127 IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INALT C',
+ ATTR,IARDF2(JARNUM,IARR),IARR)
128 IARDF2(JARLST,IARR) = K
129 CALL STORE(K,VFILE(IOFF),NF,IFLD,IRECS2(J))
130 CALL STORE(CVALS(1,J),ARRAY(IOFFCH),2,1,K)
131 CALL STORE(CVALS(2,J),ARRAY(IOFFCH),2,2,K)
132 85 CONTINUE
133 GO TO 40

C
134 90 RETURN
135 END
INPUT ROUTINES — INATTR SUBROUTINE

SUBROUTINE INATTR (ATTR, IREC, ISPEC, IRECTP)
C
C ROUTINE TO READ THE VALUE(S) OF AN ATTRIBUTE OF INTEREST
C ASSOCIATED WITH A GIVEN VEHICLE IN THE EXISTING VEHICLE FILE.
C THE ATTRIBUTE'S VALUE MAY BE A REAL, INTEGER, OR CHARACTER
C SCALAR (<= 8 CHARS) OR AN ARRAY OF REAL NUMBERS.
C THIS ROUTINE PROCESSES ATTRIBUTES WHICH WILL BE USED AS
C CONSTRAINTS AND STORED IN A RELN RECORD. SUBROUTINE INATR2
C PROCESSES THOSE THAT ARE STORED AS CONSTANTS.
C
C INPUT PARAMETERS:
C ATTR ATTRIBUTE OF INTEREST
C IREC RECORD OF IRECS CONTAINING INFORMATION ABOUT VEHICLE
C OF INTEREST
C ISPEC RECORD DESCRIBING USER SPECIFICATION
C IRECTP TYPE OF RECORD THE ATTRIBUTE IS A FIELD OF

COMMON /VEH/
DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
+ FMCNGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
+ ISENSR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),
+ IAMMOC(26,9), IENGIN(43,8), ITRANS(32,9), IFINDR(28,10),
+ IROADW(36,1C), ISPRNG(30,10), ITPACK(34,8), ISKIRT(28,8),
+ IFUEL(28,10), IFUELCL(26,9), ICROW(26,5), ICARGO(26,5),
+ IECCT(26,6), ICOMMO(28,9), IFIREX(28,10), IFENVIR(26,9),
+ IDMGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
+ IRELN(3,50), VFILE(1), IVFILE(1)
EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
+ ITURET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMCNGN(1)),
+ (MISGN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
+ IRANGR(1)), (SENSR(1), ISENSR(1)), (STABE(1), ISTABL(1)),
+ (GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
+ IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANSM(1), ITRANS(1)),
+ (FINLDR(1), IFINCR(1)), (ROADWH(1), IROADW(1)), (SPRING(1),
+ ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),

---
INPUT ROUTINES - INATTR SUBROUTINE

ISN
+ (FUEL(1), IFUEL(1), IFUELC(1)), (CREW(1),
+ ICREW(1)), (CARGO(1), ICARGO(1), (ELECTR(1), IELECTR(1)),
+ (COMMO(1), ICCOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),
+ IFENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
+ (SMOKE(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELV(1))
5 EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))
6 COMMON /VPROTO/
7 REAL*8 FLDNAM, RECNAM, DEFAULT
8 COMMON /JPARS/
9 COMMON /JPARSI/
10 COMMON /NPARS/
11 COMMON /AUX/
12 REAL*8 BLB, DASHE
13 EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
14 COMMON /ARRAYS/
15 REAL*8 MANUF, MODEL, NATION
16 DIMENSION ARRAY(1)
17 EQUIVALENCE (ARRAY(1), DETRG(1))
18 COMMON /JPARS2/
19 COMMON /SPECS/
20 REAL*8 ATRENG, ATRPER, VEHCAT, VEHCMR, VEHENG, VEHPER
21 COMMON /SPECS2/
22 REAL*8 VEH
23 REAL*8 ATT
24 DIMENSION CVALS(2,12), IVALS(12), VALS(12)
25 C MUST BACKSPACE TO POSITION RECORD POINTER APPROPRIATE FOR READING
26 BACKSPACE 2
27 C GET TYPE OF ATTRIBUTE
28 IFLD = IATTR2(ATTR, IRECTP)
29 KFLD = IFLDTP(IRECTP) + IFLD
30 ITP = IFLDTP(KFLD)
C C MARK ATTRIBUTE AS PROCESSED
INPUT ROUTINES - INATTR SUBROUTINE

ISN
IPECS(JDCONE,IREC) = 1

C GET COMPONENT IC IF THIS IS ENGINEERING PARAMETER CONSTRAINT
ID = 0
IF (IRECTP .GT. 1) ID = INDENG(ISPEC)

C TRANSLATE RELOP TO A CANONICAL VALUE FROM 1 TO 6
DO 5 J=1,NRELOP
  IF (IRECTP .EQ. 1) REL = RELPER(ISPEC)
  IF (IRECTP .GT. 1) REL = RELNAM(J)
  GO TO 8
  CONTINUE
5
  IF (IRECTP .EQ. 1) CALL ERR(26,'INATTR','VEH',ATRPER(ISPEC),+
                           RELPER(ISPEC))
  IF (IRECTP .GT. 1) CALL ERR(26,'INATTR',TYPENG(ISPEC),+
                           ATRPER(ISPEC),RELEN(I SPEC))

C KREL = IRELTP(J)

C GET THE COLUMN OF INTEREST IN THE EXISTING VEHICLE FILE
IV = IPECS(JVEH,IREC)
ICOL = ICOLS(IV)

C BRANCH ACCORDING TO CATATYPE
DO 10 I=1,NTYPES
  IF (ITYP .EQ. ITYPES(I)) GO TO 15
  CONTINUE
10
  CALL ERR(10,'INATTR',RECNAM(IRECTP),ATTR,ITYP)
  RETURN
15
GO TO (20,40,60,80),I

C READ & STORE A REAL VALUE
20 READ (2,100) (VALS(I),I=1,ICOL)
100 FORMAT (12X,12(F8.0,2X))
VALUE = VALS(ICOL)
**INPUT ROUTINES - INATTR SUBROUTINE**

C TRANSFORM VALUE IF CALLED FOR BY USER SPECIFICATION

```
51 IF (IRECTP .GT. 1) GO TO 25
52 VAL = TRANSF(VALUE, MULPER(ISPEC), VLIPER(ISPEC),
      + ADDPER(ISPEC), VL2PER(ISPEC))
53 GO TO 30
54 25 VAL = TRANSF(VALUE, MULENG(ISPEC), VLIENG(ISPEC),
      + ADDENG(ISPEC), VL2ENG(ISPEC))
55 30 TR = NEWREL(IRECTP, IFLD, KREL, VAL, ID)
56 RETURN
C
C READ & STORE AN INTEGER VALUE
```

```
57 40 READ (2, 105) (IVALS(I), I=1, ICOL)
58 105 FORMAT (12X, 12(I2, 2X))
59 VALUF = IVALS(ICOL)
C TRANSFORM VALUF IF CALLED FOR BY USER SPECIFICATION
60 IF (IRECTP .GT. 1) GO TO 45
61 IVALUE = TRANSF(VALUE, MULPER(ISPEC), VLIPER(ISPEC),
      + ADDPER(ISPEC), VL2PER(ISPEC))
62 GO TO 50
63 45 IVALUE = TRANSF(VALUE, MULENG(ISPEC), VLIENG(ISPEC),
      + ADDENG(ISPEC), VL2ENG(ISPEC))
64 50 IR = NEWREL(IRECTP, IFLD, KREL, IVALUE, ID)
65 RETURN
C
C READ & STORE AN ARRAY OF REAL VALUES
C
C DETERMINE WHICH ARRAY FILE THE INFORMATION WILL BE STORED IN
```

```
66 60 DO 62 IARR=1, NARRS
67 IF (IARDF2(IARCTP, IARR) .EQ. IRECTP .AND.
      + IARDF2(IFLDRC, IARR) .EQ. IFLD) GO TO 63
68 62 CONTINUE
69 CALL ERR(16, 'INATTR', IRECTP, IFLD, ITYP)
C
C NUMBER OF ELEMENTS IN ARRAY
```

```
70 63 NV = IARDF2(JARSIZ, IARR)
```
INPUT Routines - InAttr Subroutine

C Offset in file containing arrays
71 Ioffset = IARDF2(Jarrpt, IArr)

C Get a vacant array of type IArr
72 K = IARDF2(JARLST, IArr) + 1
73 IF (K .GT. IARDF2(JARNUM, IArr)) CALL ERR(17, 'InAttr', Attr,
+ IARDF2(JARNUM, IArr), IArr)
74 IARDF2(JARLST, IArr) = K

C Get a relation record to store the constraints in this field in
75 IR = NEWREL(Irectp, Ifld, Krel, K, ID)

C For each value of array specified in existing vehicle file,
C read, transform, & store it.
76 DO 75 I=1, NV
77 READ (2,100) (vals(J), J=1, Icol)
78 VALUE = VALS(Icol)

C Transform value if called for by user specification
79 IF (Irectp .GT. 1) GO TO 65
80 VAL = TRANSF(VALUE, MULPER(ISpec), V1PER(ISpec),
+ ADDPER(ISpec), V2PER(ISpec))
81 GO TO 70
82 65 VAL = TRANSF(VALUE, MULENG(ISpec), V1ENG(ISpec),
+ ADDENG(ISpec), V2ENG(ISpec))
83 70 CALL STOREVALVAL, ARRAY(Ioffset), NV, I, K)
84 75 CONTINUE
85 RETURN

C Read & store a character value (<= 8 chars)
86 80 READ (2,110) (cvals(1,J), cvals(2,J), J=1, Icol)
87 110 FORMAT (12X,12(2A4,2X))

C Determine which array file information will be stored in
88 DO 82 Iarr=1, Narrs
89 IF (IARDF2(Jfld, IArr) .EQ. Ifld) GO TO 83
90 82 CONTINUE
91 CALL ERR(16, 'InAttr', Irectp, Ifld, ITyp)
C
C Offset in file containing character strings
92 83 Ioffset = IARDF2(Japrept, IArr)
INPUT ROUTINES - INATTR SUBROUTINE

C GET A VACANT CHARACTER STRING RECORD
93 K = IARDF2(JARLST, IARR) + 1
94 IF (K .GT. IARDF2(JARNUM, IARR)) CALL ERR(17, 'INATTR*', ATTR,
95 + IARDF2(JARNUM, IARR), IARR)
96 IARDF2(JARLST, IARR) = K
C CHECK THAT RELATION IS EQUALITY OR INEQUALITY (OTHER RELNS NOT
C APPROPRIATE FOR CHARACTER DATA).
96 IF (KREL .NE. 3 .AND. KREL .NE. 4) CALL ERR(27, 'INATTR*',
97 + KREL, ATTR, IRECTP)
C GET A RELATION RECORD TO STORE THE CONSTRAINT ON THIS FIELD IN
97 IP = NEWREL(IRECTP, IFLD, KREL, K, ID)
C STORE CHARACTER DATA IN CHARACTER RECORD
98 CALL STORE(CVALS(1, ICOL), ARRAY(IOFFCH), 2, 1, K)
99 CALL STORE(CVALS(2, ICOL), ARRAY(IOFFCH), 2, 2, K)
100 RETURN
101 END
INPUT ROUTINES – INATR2 SUBROUTINE

SUBROUTINE INATR2 (ATTR, IREC, ISPEC, IRECTP)

C
C ROUTINE TO READ THE VALUE(S) OF AN ATTRIBUTE OF INTEREST
C ASSOCIATED WITH A GIVEN VEHICLE IN THE EXISTING VEHICLE FILE.
C THE ATTRIBUTE'S VALUE MAY BE A REAL, INTEGER, OR CHARACTER
C SCALAR (<= 8 CHARs) OR AN ARRAY OF REAL NUMBERS.
C THIS ROUTINE PROCESSES ATTRIBUTES WHICH WILL BE
C STORED AS CONSTANTS IN A COMPONENT RECORD. SUBROUTINE INATTR
C PROCESSES THOSE THAT ARE STORED AS CONSTRAINTS.
C
C INPUT PARAMETERS:
C ATTR ATTRIBUTE OF INTEREST
C IREC RECORD OF IRECS CONTAINING INFORMATION ABOUT VEHICLE
C OF INTEREST
C ISPEC RECORD DESCRIBING USER SPECIFICATION
C IRECTP TYPE OF RECORD THE ATTRIBUTE IS A FIELD OF
C
COMMON /VEH/
DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
  + FMAGN(40,10), FMIGN(40,7), IAUDGUN(40,6), IRANGR(28,7),
  + ISENSE(32,9), INSTAB(3,6), IGMTL(28,9), IAMMO(3,33),
  + IAMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),
  + IRODH(36,10), ISPRNG(30,10), ITRAKT(34,8), ISKIRT(28,8),
  + IUEL(28,10), IFEULC(26,9), ICREW(26,5), ICARGO(26,5),
  + ITEC(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),
  + IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
  + IRELN(3,50), IVECLE(1), IVECLE(1)
EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURET(1),
  + ITURET(1)), (MAING(1), FMAING(1)), (MACGN(1), FMAGN(1)),
  + FMSGN(1), FMIGN(1), (AUDGUN(1), IADGUN(1)), (RANGR(1),
  + IRANGR(1)), (SENSE(1), ISENSE(1)), (STAB(1), INSTAB(1)),
  + (GNC(1), IGM(1)), (AMMIO(1), IAMMO(1)), (AMMOC(1),
  + IAMOC(1), IENGNE(1), IENG(1)), (TRANSM(1), ITRANS(1)),
  + (FINDR(1), IFINDR(1)), (ROADW(1), IRODH(1)), (SPRING(1),
  + ISPRING(1)), (TRAK(1), ITRAK(1)), (SKIRT(1), ISKIRT(1)),

C-33
C -34

INPUT ROUTINES – INATR2 SUBROUTINE

ISN

+ (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
+ ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
+ (COMM(1), ICOMM(1)), (FIREX(1), IFIREX(1)), (ENVR(1),
+ IENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
+ (SMOKEG(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))

EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))
COMMON /VPROTO/
REAL*8 FLDNAM, RECNAM, DEFAULT
COMMON /JPARS/
COMMON /JPARS1/
COMMON /NPARS/
COMMON /AUX/
REAL*8 BLS, CASHES
EQUIVALENCE (FNULL, NULL), (IBLS, BL5)
COMMON /ARRAYS/
REAL*8 MANUF, MODEL, NATION
DIMENSION ARRAY(1)
EQUIVALENCE (APRAY(1), DETRNG(1))
COMMON /JPARS2/
COMMON /SPFCS/
REAL*8 ATRENG, APRPER, VEHCAT, VEHCMAP, VEHEW, VEHPER
COMMON /SPFCS2/
REAL*8 VEHS

REAL*8 ATTR
DIMENSION CVALS(2, 12), IVALS(12), VALS(12)

C MUST BACKSPACE TO POSITION RECORD POINTER APPROPRIATE FOR READING
BACKSPACE 2

C GET TYPE OF ATTRIBUTE
IFLD = IATR2(ATTR, IRECTP)
KFLD = IFLDPT(IRECTP) + IFLD
ITYP = IFLDTYP(KFLD)

C GET COMPONENT IC AND COMPONENT TYPE
INPUT ROUTINES -  INATR2 SUBROUTINE

49 100 FORMAT (12X,12(F8.0,2X))

C STORE VALUE IN THE RECORD
50 CALL STORE(VALS(ICOL),VFILE(IOFF),NF,IFLD,IR)
51 RETURN

C READ & STORE AN INTEGER VALUE
52 40  READ (2,105) (IVAL$(I),I=1,ICOL)
53 105 FORMAT (12X,12(I8,2X))
54 CALL STORE(IVAL$(ICOL),VFILE(IOFF),NF,IFLD,IR)
55 RETURN

C READ & STORE AN ARRAY OF REAL VALUES
C DETERMINE WHICH ARRAY FILE THE INFORMATION WILL BE STORED IN
56 60  DO 62 IARR=1,NARRS
57 IF (IARDF2(IJRECTP,IARR) .EQ. IRECTP .AND. 
     + IARDF2(JFLDRC,IARR) .EQ. IFLD) GO TO 63
58 62  CONTINUE
59 CALL ERR(16,'INATR2',IRECTP,IFLD,ITYP)

C NUMBER OF ELEMENTS IN ARRAY
60 63  NV = IARDF2(JARSIZ,IARR)
C OFFSET IN FILE CONTAINING ARRAYS
61 IOFFAR = IARDF2(JAPRPT,IARR)
C GET A VACANT ARRAY OF TYPE IARR
62 K = IARDF2(JARLST,IARR) + 1
63 IF (K.GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INATR2',Attr,
     + IARDF2(JARNUM,IARR),IARR)
64 IARDF2(JARLST,IARR) = K
C STOP THE POINTER TO THIS ARRAY
65 CALL STORE(K,VFILE(IOFF),NF,IFLD,IR)
C FOR EACH VALUE OF ARRAY SPECIFIED IN EXISTING VEHICLE FILE,
C READ, TRANSFORM, & STORE IT.
66  DO 75 I=1,NV
67  READ (2,100) (VALS(J),J=1,ICOL)
68  CALL STORE(VALS(ICOL),ARRAY(IOFFAR),NV,I,K)
**INPUT ROUTINES - INATR2 SUBROUTINE**

```fortran
ISN
69    75    CONTINUE
70    RETURN

C READ & STORE A CHARACTER VALUE (<= 8 CHARS)
71    80    READ (2,110) (CVALS(I,J),ICOL)
72    110    FORMAT (12X,12(2A4,2X))

C DETERMINE WHICH ARRAY FILE INFORMATION WILL BE STORED IN
73    DO 82 IARR=1,NARRS
74    IF (IAPDF2(JFLDPC,IARR) .EQ. IFLD) GO TO 83
75    82    CONTINUE
76    CALL ERR(16,'INATR2',IRECTP,IFLD,ITYP)

C OFFSET IN FILE CONTAINING CHARACTER STRINGS
77    83    JOFFCH = IARDF2(JARRPT,IARR)
78    C GET A VACANT CHARACTER STRING RECORD
79    K = IARDF2(JARLST,IARR) + 1
80    IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INATR2',Attr,
81    + IARDF2(JARNUM,IARR),IARR)
82    IARDF2(JARLST,IARR) = K

C STORE POINTER TO THE CHARACTER RECORD
83    CALL STORE(K,VFILE(JOFFCH),NF,IFLD,IR)

C STORE CHARACTER DATA IN CHARACTER RECORD
84    CALL STORE(CVALS(1,ICOL),ARRAY(JOFFCH),2,1,K)
85    CALL STORE(CVALS(2,ICOL),ARRAY(JOFFCH),2,2,K)
86    RETURN
87    END
```
INPUT ROUTINES - INFNC SUBROUTINE

1 SUBROUTINE INFNC

SUBROUTINE READS A FORMATTED FILE OF DATA VARIABLE
HEADERS FOLLOWED BY DATA FOR EACH VARIABLE. THE DATA VARIABLES
CONSIST OF SEQUENTIAL VARIABLES IN A COMMON BLOCK. EACH VARIABLE
HEADER RECORD HAS THE FOLLOWING FIELDS:

<table>
<thead>
<tr>
<th>FIELD</th>
<th>NAME</th>
<th>COL 1-8</th>
<th>DIM 1</th>
<th>DIM 2</th>
<th>DIM 3</th>
<th>STAR</th>
<th>RED. DIM 1</th>
<th>REPL 1</th>
<th>RED. DIM 2</th>
<th>REPL 2</th>
<th>RED. DIM 3</th>
<th>REPL 3</th>
<th>FTN FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>COL 1-8</td>
<td></td>
<td>9-11</td>
<td>12-14</td>
<td>15-17</td>
<td>19</td>
<td>20-22</td>
<td>23</td>
<td>24-25</td>
<td>26</td>
<td>27-28</td>
<td>29</td>
<td>31-78</td>
</tr>
<tr>
<td>DIM 1</td>
<td>1ST DIMENSION OF VARIABLE IN PROGRAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIM 2</td>
<td>2ND DIMENSION OF VARIABLE IN PROGRAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIM 3</td>
<td>3RD DIMENSION OF VARIABLE IN PROGRAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STAR</td>
<td>'*' IF DATA FOR VARIABLE IS ALL ZEROS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RED. DIM 1</td>
<td>(OPTIONAL) REDUCED DIMENSION APPLYING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPL 1</td>
<td>TO DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPL 2</td>
<td>(OPTIONAL) DATA GIVEN ALONG FIRST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPL 3</td>
<td>SECTION OF DIMENSION IS TO BE REPLICATED ALONG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RED. DIM 2</td>
<td>LATER SECTIONS IF FIELD CONTAINS &quot;#&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPL 3</td>
<td>(OPTIONAL) REDUCTION OF THE DIMENSION 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RED. DIM 3</td>
<td>(OPTIONAL) REPLICATION SYMBOL (#)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPL 3</td>
<td>(OPTIONAL) REPLICATION FACTOR 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTN FORMAT</td>
<td>FORTRAN FORMAT (IN PARENTHESES)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

THE DATA VALUES FOR A VARIABLE FOLLOW THE HEADER CARD. FOR
ARRAYS THE NUMBER OF DATA VALUES EXPECTED IS THE PRODUCT OF THE
NON-ZERO DIMENSIONS GIVEN ON THE HEADER CARD. IF REDUCED DIMENSIONS
ARE SPECIFIED THEY ARE USED INSTEAD OF THE ACTUAL DIMENSIONS IN
COMPUTING THE NUMBER OF EXPECTED DATA VALUES. FOR SCALARS, VARIABLES
WITH NO DIMENSIONS, A SINGLE VALUE IS EXPECTED. IF A '*' IS GIVEN
IN THE STAR FIELD, ZERO DATA VALUES ARE EXPECTED & THE NEXT HEADER
CARD SHOULD FOLLOW IMMEDIATELY. THE FORMAT THAT THE DATA VALUES
APPEAR IN IS UP TO THE USER. THE FORMAT IS INDICATED ON THE HEADER
CARD & MUST BE A VALID FORTRAN FORMAT SPECIFICATION.
A BLOCK OF DATA VARIABLES MAY BE TERMINATED BY A RECORD WITH
"ENDBLK" IN COLS 1-6. ALL VARIABLES WITHIN A BLOCK MUST BE IN PROPER
INPUT ROUTINES - INFUNC SUBROUTINE

C SEQUENCE & IN 1:1 CORRESPONDENCE WITH THE VARIABLES OF THE ASSOCIATED
C COMMON BLOCK USED IN THE MAIN PROGRAM. SEE THE USER'S MANUAL FOR A
C LIST OF THE INPUT VARIABLES EXPECTED, THEIR ORDER, & THEIR
C DIMENSIONS.
C THE DATA SET REFERENCE NUMBERS USED ARE AS FOLLOWS:
C 1 INPUT
C 8 OUTPUT ECHO OF DATA FILE HEADER CARDS

2 COMMON /DATA /
3 COMMON /DATA2 /
4 EQUIVALENCE (VEC(1), ACCEL(1))

C REAL VEC(2), W(100)
6 REAL*8 VAR, ENDBLK, ENDFIL
7 INTEGER FMT(12)
8 DATA STAR1/I1*#, ENDBLK/6HENDBLK/, ENDFIL/6HENDFIL/
9 DATA SHARP/I1*#

C CALL FTNMDI('SET MODECHECK=OFF', 17)
11 10 JJ0 = 0
12 11 READ (1, 100, END=44) VAR, I, J, K, STAR, I2, S1, J2, S2, K2, S3, FMT
13 WRITE (8, 101) JJ0, VAR, I, J, K, STAR, I2, S1, J2, S2, K2, S3, FMT
14 100 FORMAT (A8, 3I3, 1X, A1, I3, 2(A1, I2), A1, 1X, 12A4)
15 101 FORMAT (I6, 2X, A8, 3I3, 1X, A1, I3, 2(A1, I2), A1, 1X, 12A4)
16 IF (VAR *EQ. ENDBLK *OR. VAR *EQ. ENDFIL) GO TO 44
17 JSIZE = 1
18 IF (I .NE. 0) JSIZE = JSIZE*I
19 IF (J .NE. 0) JSIZE = JSIZE*J
20 IF (K .NE. 0) JSIZE = JSIZE*K
21 IF (JSIZE .GT. 5000) STOP 100
22 JJ2 = JJ0 + JSIZE
23 IF (JJ2 .GT. 5000) STOP 200
24 JJ1 = JJ0 + 1
25 14 IF (STAR .EQ. STAR1) GO TO 40
C *** IF ALL ZEROS, THEN SKIP THIS VARIABLE
26 IF (I .NE. 0) GO TO 31
**INPUT ROUTINES - INFUNC SUBROUTINE**

C *** IS THIS A SCALAR
27 READ (1,FMT) VEC(JJ1)
29 GO TO 40

C 31 IF (J .NE. 0) GO TO 32
C *** IS IT A VECTOR ?
30 IF (I2 .EQ. 0) I2 = 1
31 JJ3 = JJ0 + I2
32 READ (1,FMT) (VEC(M), M=JJ1, JJ3)
33 GO TO 40

C 34 32 IF (K .NE. 0) GO TO 36
C *** IS IT A 2-D ARRAY
35 IF (I2 .NE. 0) GO TO 33
36 READ (1,FMT) (V(M), M=1, IJ)
37 DO 34 II=1, I2
38 34 J1 = J1 - 1
39 M1 = JJ0 + II + I * J1
40 M2 = II + I2 * J1
41 DO (VEC(M1) = V(M2)
42 34 GO TO 40

C 47 36 IF (I2 .NE. 0) GC TO 38
48 READ (1,FMT) (VEC(M), M=JJ1, JJ2)
49 GO TO 40
51 38 IJK = I2 * J2 * K2
52 READ (1,FMT) (V(M), M=1, IJK)
53 DO 39 II = 1, I2
54 39 J1 = J1 - 1
55 DO 39 K1 = 1, K2
56 K11 = K1 - 1
INPUT ROUTINES — INFUNC SUBROUTINE

\[ \begin{align*}
M1 &= J0 + I1 + I1 + [J1 + 1]*K11 \\
M2 &= I1 + I2 + J1 + J2 + 2*K11 \\
\text{VEC}(M1) &= V(M2)
\end{align*} \]

IR = 1
IF (S1 .EQ. SHARP) IREP = IREP+1
IF (S2 .EQ. SHARP) IREP = IREP+J
IF (S3 .EQ. SHARP) IREP = IREP+K
J2SIZE = JSIZE/IREP
L1 = J0
DO 42 ICT = 1, IREP
L1 = L1 + J2SIZE
42 GO TO 11
L1 = J0 + J2SIZE
VECM1 = VEC(J0 + L2)
INPLT ROUTINES - INIT SUBROUTINE

SUBROUTINE INIT
ROUTINE TO INITIALIZE VARIABLES AT BEGINNING OF RUN.

COMMON /VEH/
DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
+FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
+ISENSR(32,9),ISTABL(30,6),IGNCTR(28,9),IAMNO(30,30),
+IAMMOC(26,9),IENGIN(45,9),ITRNS(32,8),IFINDR(28,10),
+IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
+IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
+IFECV(26,6),ICOMMO(28,9),IFIREX(28,10),IVENIR(26,9),
+IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IIEWSY(26,10),
+IRELN(3,50),IVFILE(1),IVFILE(1)
EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
+ITURET(1)),(MAING(1),FMAING(1)),(MACGN(1),FMACGN(1)),
+(MIISGN(1),FMISGN(1)),(IADGUN(1),IADGUN(1)),(RANGR(1),
+ISENSR(1),ISENSR(1)),(STABL(1),ISTABL(1)),
+(IGNCTR(1),IGNCTR(1)),(IAMNO(1),IAMNO(1)),(IAMMOC(1),
+IENGIN(1),IENGIN(1)),(ITRNS(1),ITRNS(1)),
+(IFINDR(1),IFINDR(1)),(IROADW(1),IROADW(1)),(ISPRNG(1),
+ITRACK(1),ITRACK(1)),(ISKIRT(1),ISKIRT(1)),
+(IFUEL(1),IFUELC(1)),(IFUELC(1),IFUEL(1)),(ICREW(1),
+ICCREW(1)),(ICARGO(1),ICARGO(1)),(IFECV(1),IFECV(1)),
+(IFIREX(1),IFIREX(1)),(IVENIR(1),IVENIR(1)),(IDIAGN(1),IDIAGN(1),
+ISIGSP(1),ISIGSP(1)),
+ISMOKE(1),ISMOKE(1),IIEWSY(1),IIEWSY(1),IRELN(1),IRELN(1))
EQUIVALENCE (VEHICLE(1),IVFILE(1),IVFILE(1))

COMMON /VPPROTO/
REAL*8 FLDNAM,RECNAM,DFALT
COMMON /NPARS/
COMMON /OUTPUT/
LOGICAL*1 DF
REAL*8 XNAT,XMANUF,XMODEL
COMMON /AUX/
REAL*8 BL8,DASHES
INPUT ROUTINES - INIT SUBROUTINE

14 EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
15 COMMON /ARRAYS/
16 REAL*8 MANUF, MODEL, NATION
17 DIMENSION ARRAY(I)
18 EQUIVALENCE (ARRAY(I), DETRNG(I))
19 COMMON /JPARS2/
20 COMMON /SPECS/
21 REAL*8 ATRENG, ATRPER, VEHCAT, VEHCP, VEHENG, VEHPER

C INITIALIZE IFLOPT BASED ON NUMBER OF FIELDS IN EACH TYPE OF
C RECORD
C
22 N = 0
23 NN = 1
24 IFLOPT(I) = N
25 DO 10 I = 1, NRECTP
26 N = N + NFLDS(I)
27 IFILPT(I) = NN
28 NN = NN + NFLDS(I) * NRECS(I)
29 10 IFLOPT(I+1) = N
C
C DETERMINE NUMBER OF CELLS IN ALL RECORDS COMBINED
C
30 NCELL = 0
31 DO 15 I = 1, NRECTP
32 NCELL = NCELL + NFLDS(I) * NRECS(I)
33 15
C
C INITIALIZE THE FIELDS OF ALL RECORDS TO NULL
C
34 CALL NULIFY(VEHICLE, NCELL)
C
C INITIALIZE VARIABLES THAT WILL CONTAIN TEXT TO BLANKS
C
35 CALL INITVL(FLEXAM, NFLDRC*2, BLS)
36 CALL INITVL(RECAAM, NRECTP*2, BLS)
37 CALL INITVL(CMPCD0, NCMP*TP, BLS)
INPUT ROUTINES - INIT SUBROUTINE

ISN
37    CALL INITVL(UNITS,NFLDRC,BLS)
C
C INITIALIZE COUNTERS TO ZERO
C
38    LSTCMP = 0
39    LSTENG = 0
40    LSTPER = 0
41    CALL IZERO(LSTREC,NRECTP)
42    CALL IZERO(MAXNUM,NCMPTP)
43    CALL IZERO(MINNUM,NCMPTP)
44    CALL IZERO(NALTCS,NCMPTP)
C
C INITIALIZE FIELD TYPES AND VALUES TO NULL
C
45    CALL NULIFY(IFLDTP,NFLDRC)
46    CALL NULIFY(IFLDVL,NFLDRC)
47    CALL NULIFY(ICMPPT,NCMPTP)
C
C INITIALIZE POINTERS TO ARRAYS
C
48    NN = 1
49    DO 20 I=1,NARRS
50       IARDF2(JARLST,I) = 0
51       IARDF2(JAPRPT,I) = NN
52     NA = IARDF2(JARSIZE,I) * IARDF2(JARNUM,I)
53    20       NN = NN + NA
C
54    RETURN
55    END
INPUT ROUTINES - INPRT SUBROUTINE

SUBROUTINE INPRT

C ROUTINE FOR READING THE PROTOTYPE VEHICLE DESCRIPTION FILE AND
C STORING THIS INFORMATION IN INTERNAL RECORDS AND ARRAYS.

COMMON /VPRT0/
REAL*8 FLDNAH,RECNAM,DFALT
COMMON /NPARS/
COMMON /OUTPUT/
LOGICAL*1 DF
REAL*8 XNAT,XMANUF,XMODEL
COMMON /AUX/
REAL*8 BL8,DASHES
EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
COMMON /ARAYS/
REAL*8 MANUF,MODEL,NATION
DIMENSION ARRAY(1)
EQUIVALENCE (ARRAY(I),DETRNG(I))
COMMON /JPARS2/
COMMON /SPECS/
REAL*8 ATRENG,ATRPER,VEHCAT,VEHMP,VEHENG,VEHPER

COMMON //
DIMENSION CONAME(7)
EQUIVALENCE (CMPTYP,CNAME(I)), (CMPTYP,CMPTP2)
REAL*8 V,VH,CMPTYP,WHNAME,PPARS,COMAMT,COMEND,DFVAL
DATA STAR/***/,'VEHICLE'//'DATA KPA/**RA/**,PPARS/*1. PERFO*/,COMAMT'/',A. COMM',/
+  'COMEND'/','SPEC'/','KCB/*CR'/'

C INITIALIZATIONS
LSTOUT = 0
IARRAY = 0
ICOM = 0
KMP=0
INPUT ROUTINES - INPROT SUBROUTINE

ISN
28       K = 10
29       LARDEF = 0
30       LEVEL = 0

C
C READ VEHICLE HEADER; SKIP OVER COMMENT LINES
31      10   READ (3,100) C,V,VEHCAT
32      100  FORMAT (A1,A7,3X,A6)
33      IF (C .EQ. STAR) GO TO 10
34      IF (V .NE. VH) CALL ERR(1,'INPROT',V,0,0)
35      IREC = 1
36      IRECPT = IFLCPT(IREC)
37      RECNAM(IREC) = VH
38      ISTATE = 1

C
C READ A NEW RECORD
39      15   READ(3,105,END=90) C,CMPTYP,FILL,CCOD,ITYPE
40      105  FORMAT(A1,A8,5A4,23X,A2)

C
C IS THIS A COMMENT? IF SO, SKIP INPUT DATA PROCESSING.
C
41      IF(C.EQ.STAR) GO TO 30

C
C IF THIS IS DATA, BACKSPACE AND READ INPUT RECORD AGAIN.
C
42      BACKSPACE 3

C
C READ DATA ON APPROPRIATE FORMAT AS INDICATED BY ITYPE;
C ITYPE.EQ.'C8' IMPLIES CHARACTER DFVAL, 'NE.'C8' IMPLIES NUMERIC.
C
43      IF(ITYPE.EQ.'K8') READ(3,107) CCOD,RWNAME,IFLOV,MIN,MAX,UNIT,
+       DFVAL
44      107   FORMAT(25X,A4,A8,10X,12,5X,213,2X,A4,2X,A8)
45      IF(ITYPE.NE.'K8') READ(3,109) CCOD,RWNAME,IFLOV,MIN,4AX,UNIT,
+       DFVAL
46      109   FORMAT(25X,A4,A8,10X,12,5X,213,2X,A4,2X,F8.0)
INPUT ROUTINES - INPROT SUBROUTINE

ISN
C CHECK FOR NEW COMPONENT, INDICATED BY BLANK RWNAME AND IFLOV.
C IF TRUE, SKIP TO NEW COMPONENT SECTION.
47
C IF (RWNAME .EQ. BLR .AND. IFLOV .EQ. 0) GO TO 20
C
C PROCESS NEW ATTRIBUTE OF CURRENT RECORD
48
IF (IFLOV .GT. NFDOS(IREC)) CALL ERR(2, 'INPROT', RWNAME, IFLOV, IREC)
49
IFLDRG = IRECPT + IFLOV
50
FLDNAM(IFLDRG) = RWNAME
51
IFLOVL(IFLDRG) = IFLOV
52
IFLOTP(IFLDRG) = ITYPE
53
UNITS(IFLDRG) = UNIT
54
DFVALT(IFLDRG) = DFVAL
55
IF (ISTATE .NE. 1) GO TO 18
56
MINNMIN(IFLOV) = MIN
57
MAXIMUM(IFLOV) = MAX
58
KMP = KMP + 1
59
KOUT(KMP) = IFLOV
60
18
IF (ISTATE .EQ. 3 .OR. ICOM .EQ. 0) CALL STONAM(DNAME, IFLDRG)
61
IARRAY = 0
62
IF (ITYPE .EQ. KRA) IARRAY = 1
63
GO TO 15
C
C PROCESS A NEW COMPONENT RECORD
64
20
IPEC = IREC + 1
65
ICMP = IREC - 1
66
IF (IREC .GE. NRECP) CALL ERR(3, 'INPROT', IREC, NRECP, 0)
67
IRECPT = IFLOPT(IREC)
68
RECNAM(IREC) = CMPTYP
69
CMPCOD(ICMP) = CCOD
70
CALL STONAM(CNAME, 0)
71
GO TO 15
C
72
30
CONTINUE
C
C COMMENT PROCESSING SECTION
INPUT ROUTINES — INPROT SUBROUTINE

C END-OF-ARRAY
C
89 IARRAY = 0
C
90 38 CONTINUE
C
C PROCESSING FOR COMMENTS OTHER THAN ARRAY LABELS
C
C IS THIS THE DEFINITION OF AN ATTRIBUTE LEVEL? IF SO SKIP IT
91 IF (CMPTYP .EQ. BLA) GO TO 15
C
C BRANCH TO CODE ACCORDING TO STATE
C STATE = 1: READING VEHICLE COMPONENT HEADERS; CHECK FOR
C BEGINNING OF PERFORMANCE PARAMETERS
C STATE = 2: READING PERFORMANCE PARS; CHECK FOR END OF PERF PARS
C STATE = 3: READING COMPONENTS & THEIR ATTRIBUTES; CHECK FOR
C BEGINNING OF COMMON ATTRIBUTES
C STATE = 4: READING COMMON ATTRIBUTES; CHECK WHETHER HAVE
C REACHED END OF SECTION OR HAVE ALREADY READ THEM
C
92 GO TO (40, 50, 60, 70), ISTATE
C
C STATE 1. IS THIS BEGINNING OF PERFORMANCE PARS?
93 40 IF (CMPTYP .NE. PPARS) GO TO 45
94 IPPARS = LSTOUT + 1
95 ISTATE = 2
96 GO TO 15
C STORE NAME IN OUTNAM
97 45 CALL STONAM(CNAME, 0)
98 GO TO 15
C
C STATE 2. IS THIS THE END OF THE PERFORMANCE PARS?
99 50 IF (CMPTYP .NE. DASHES) GO TO 55
100 TCMTATR = LSTCUT + 1
101 ISTATE = 3
102 GO TO 15
INPUT ROUTINES - INPROT SUBROUTINE

ISN

C
C CHECK WHETHER CURRENTLY PROCESSING ARRAY LABELS; IF NOT SKIP
C ARRAY LABEL SECTION
73 IF (IARRAY .NE. 1) GO TO 38
C
C ARRAY LABEL PROCESSING SECTION
C
C CHECK FOR END OF CURRENT ARRAY LABELS; IF TRUE,
C SKIP TO END-OF-ARRAY
74 IF (CMPTP2 .NE. BL5 OR .FILL(1) .EQ. BL5) GO TO 37
C
C IF THIS IS LABEL OF ANOTHER LEVEL, STORE IT
75 32 IF (CMPTYP .NE. BJ8) GO TO 34
76 LEVEL = LEVEL + 1
77 IF (LEVEL .GT. NNLARLAB) CALL ERR(36,'INPROT',IREC,IFLDV,LEVEL)
78 ILEVEL = ILEVEL + 1
79 CALL COPYONAME(3),ARRLAB(1,LEVEL),5
80 GO TO 15
C
C THIS IS A NEW DIMENSION NAME. CHECK THAT PREVIOUS DIMENSION
C HAD RIGHT NUMBER OF LEVELS
81 34 IF (LARDEP .EQ. 0) GO TO 36
82 IF (ILEVEL .NE. IARDEF(JNLEVEL,LARDEP)) CALL ERR(34,'INPROT',
   + IREC,IFLDV,ILEVEL)
C
C CHECK THAT CURRENT DIMENSION IS DEFINED IN IARDEF
83 36 LARDEP = LARDEP + 1
84 IF (IARDEF(IARCTP,LARDEP) .NE. IREC .OR. IARDEF(JFDDC,LARDEP)
   + .NE. IFLDV) CALL ERR(35,'INPROT',IREC,IFLDV,LARDEP)
85 ILEVEL = 0
C STORE POINTER TO LEVEL
86 IARDEF(JLEVEL,LARDEP) = LEVEL + 1
87 GO TO 15
C
88 37 CONTINUE
C
INPUT ROUTINES - INPROT SUBROUTINE

C SAVE NAME IN OUTNAM
103  55  CALL STONAM(CNAME,0)
104  GO TO 15
C
C STATE 3. IS THIS THE BEGINNING OF COMMON ATTRIBUTES?
105  60  IF (CMPTYP .NE. COMATR) GO TO 65
106       ISTATE = 4
107  GO TO 15
C
C HAS THE COMMON ATTRIBUTE SECTION BEEN PROCESSED YET?
108  65  IF (ICOM .NE. 0) GO TO 68
C IF NOT, STORE THESE FIRST FEW LINES IN A TEMPORARY LOCATION
C AND LATER MOVE THEM TO THE END OF THE COMMON ATTRIBUTE SECTION.
109       CALL COPY(CNAME,OUTNAM(1,NOUT-K),NOUTWO)
110  K = K - 1
111  GO TO 15
C
C STORE NAME IN OUTNAM
112  68  IF (CMPTYP .EQ. CASHES) GO TO 15
113       CALL STONAM(CNAME,0)
114  GO TO 15
C
C STATE 4. HAS ONE ALREADY READ THESE ATTRIBUTES ONCE?
115  70  CONTINUE
C IS THIS THE END OF THE SECTION OF COMMON ATTRIBUTES?
116  IF (CMPTYP .NE. COMEND .AND. CMPTYP .NE. DASHES) GO TO 75
117       ISTATE = 3
118  IF (ICOM .NE. 0) GO TO 15
C
C AT THIS POINT ALL COMMENTS ASSOCIATED WITH COMMON ATTRIBUTES
C HAVE BEEN READ ONCE; WE SET POINTER AND FLAGS TO INDICATE THIS
C
119       IEPARS = LSTOUT + 1
120       ICOM = 1
C INSERT THE TEMPORARILY STORED NAMES INTO THE LIST AT THIS POINT
121       NR = 10 - K
INPUT ROUTINES - INPROT SUBROUTINE

122   CALL COPY(OUTNAM(1,NOUT-1),OUTNAM(1,LSTOUT+1),NR*NOUTWO)
123   CALL IZERO(OUTF(LSTOUT+1),NR)
124   LSTOUT = LSTOUT + NR
125   GO TO 15

C STORE NAME IN OUTNAM

126   75   IF(ICOM.EQ.0)CALL STCNAM(ONAME,0)
127       GO TO 15

C

128   90   RETURN
129       END
INPUT ROUTINES - INSPEC SUBROUTINE

1 SUBROUTINE INSPEC

2 COMMON /NPARS/
3 COMMON /AUX/
4 REAL*8 BL8,DASHES
5 EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
6 COMMON /SPECs/
7 REAL*8 ATRENC,ATRPER,VEHCAT,VEHCFM,VEHHG,VEHPER

8 DIMENSION CMPHD(4)
9 REAL*8 V
10 DATA VH/*VEHI*/,WT/*WEIG*/,GWT/*GROS*/,EVAL/*EVAL*/,
+ DESIR/*DESI*/,CMPHD/*C','E','P','V*/,VEH/*VEH*/

C READ & ECHO VEHICLE CLASS
11 READ (5,100) HEAD,V
12 WRITE (8,100) HEAD,V
13 100 FORMAT (A4,16X,A8)
14 IF (HEAD .NE. VH) CALL ERR(11,'INSPEC',HEAD,VH,0)
15 IF (V .NE. VEHCAT) CALL ERR(12,'INSPEC',V,VEHCAT,0)

C READ & ECHO GROSS WEIGHT ESTIMATE FOR VEHICLE
16 READ (5,105) HEAD,VEHWT
17 WRITE (8,105) HEAD,VEHWT
18 105 FORMAT (A4,16X,F8.0)
19 IF (HEAD .NE. WT .AND. HEAD .NE. GWT) CALL ERR(11,'INSPEC',
+ HEAD,WT,0)

C READ & ECHO SET OF EVALUATION WEIGHTS FOR VARIOUS CATEGORIES
20 READ (5,110) HEAD,VEHEVL
21 WRITE (8,110) HEAD,VEHEVL
22 110 FORMAT (A4,16X,5F8.0)
INPUT ROUTINES - INSPEC SUBROUTINE

23 IF (HEAD .NE. EVAL .AND. HEAD .NE. DESIR) CALL ERR(11,'INSPEC',
   + HEAD,EVAL,0)

C
C READ HEADER
24 READ (5,115,END=90) HEAD
25 115 FORMAT (A1)
C
C BRANCH ACCORDING TO HEADER
26 DO 10 I=1,4
27 IF (CMPHD(I) .EQ. HEAD) GO TO (20,30,40,40),I
28 10 CONTINUE
29 CALL ERR(14,'INSPEC',HEAD,0)
C
C READ & STORE COMPONENT SPECIFICATIONS
30 READ (5,115,END=90) HEAD
31 IF (HEAD .NE. BLS) GO TO 5
32 BACKSPACE 5
33 I = LSTCMP + 1
34 IF (I .GT. NCSPCS) CALL ERR(13,'INSPEC','COMPONENT',NCSPCS,0)
35 READ (5,120,END=90) HEAD,NUMCMP(I),TYPCMP(I),INDCMP(I),VEHCP(I)
36 WRITE (8,120) HEAD,NUMCMP(I),TYPCMP(I),INDCMP(I),VEHCP(I)
37 120 FORMAT (A1,12,1X,A3,1X,12,1X,46)
38 LSTCMP = I
39 GO TO 20
C
C READ & STORE ENGINEERING PARAMETER SPECIFICATIONS
40 I = LSTENG + 1
41 IF (I .GT. NESPSCS) CALL ERR(13,'INSPEC','ENG PAR',NESPSCS,0)
42 READ (5,125,END=90) HEAD,TYPENG(I),ATRENG(I),RELENG(I),VEHENG(I),
   + MULENG(I),V1ENG(I),ADDENG(I),VL2ENG(I),INDENG(I)
43 WRITE (8,125) HEAD,TYPENG(I),ATRENG(I),RELENG(I),VEHENG(I),
   + MULENG(I),V1ENG(I),ADDENG(I),VL2ENG(I),INDENG(I)
44 125 FORMAT (A1,A3,1X,A8,1X,A2,1X,A6,1X,A1,1X,F8.0,1X,A1,1X,F8.0,1X,12)
45 IF (HEAD .NE. BLS) GO TO 5
46 LSTENG = I
47 GO TO 30
INPUT ROUTINES - INSPEC SUBROUTINE

C
C. READ & STORE PERFORMANCE PARAMETERS

48 40  I = LSTPER + 1
49  IF (I .GT. NPSPCS) CALL ERR(13,'INSPEC','PERF PAR',NPSPCS,0)
50       READ (5,130,END=93) HEAD,TYPPER,ATRPER(I),RelpER(I),
51          +   VEHPER(I),MULPER(I),VLI PER(I),ADDPER(I),VL2PER(I)
52 130  FORMAT (A1,A3,1X,A8,1X,A2,1X,A6,1X,A1,1X,F8.0,1X,A1,1X,F8.0)
53  IF (HEAd .NE. BLS) GO TO 5
54  IF (TYPPER .NE. BLS .AND. TYPPER .NE. VEH) CALL ERR(15,'INSPEC',
55              +   TYPPER,0,0)
56       LSTPER = I
57  GO TO 40
58  LARGE RETURN
59  END
INPUT ROUTINES - INVEHC SUBROUTINE

SUBROUTINE INVEHC

ROUTINE TO PRODUCE INTERNAL VEHICLE SPECIFICATIONS FROM THE USER'S INPUT SPECIFICATIONS AND A FILE OF EXISTING VEHICLES.

COMMON /VEH/
DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
+ FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
+ ISENSR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),
+ IAMMOC(26,9), IENGIN(40,8), ITANS(32,8), IFINDR(28,10),
+ IPOADW(36,10), ISPRNG(33,10), ITRACK(34,8), ISKRT(28,8),
+ IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),
+ IELECT(26,6), ICCMNO(28,9), IFIREX(28,10), IENVIR(26,9),
+ IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
+ IRELN(3,50), VFILE(1), IVFILE(1)

EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURET(1),
+ ITURET(1)), (MAINGN(1), FMAING(1)), (MACGN(1), FMACGN(1)),
+ (MISGN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
+ IRANGR(1)), (SENSOR(1), ISENSR(1)), (STABLE(1), ISTABL(1),
+ (UNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
+ IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANS(1), ITANS(1)),
+ (FINDR(1), IFINDR(1)), (ROADW(1), IPOADW(1)), (SPRNG(1),
+ ISPONG(1)), (TRACK(1), ITRACK(1)), (SKIP(1), ISKRT(1)),
+ (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
+ ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
+ (COMM(1), ICCMNO(1)), (FIREX(1), IFIREX(1)), (ENVIR(1),
+ IENVIR(1)), (DIAGN(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
+ (SMOKE(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))

EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))

COMMON /VPROTO/
REAL*8 FLDNA, RECNA, Default
COMMON /JPARt/
COMMON /JPARS/
COMMON /NPARS/
COMMON /AUX/
INPUT ROUTINES - INVEHC SUBROUTINE

ISN
12 REAL*8 BL8,DASHES
13 EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
14 COMMON /ARRAYS/
15 REAL*8 MANUF,MODEL,NATION
16 DIMENSION ARRAY(1)
17 EQUIVALENCE (ARRAY(1),DETRNG(I))
18 COMMON /JPARS2/
19 COMMON /SPECS/
20 REAL*8 ATRFNG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
21 COMMON /SPECS2/
22 REAL*8 VEHS

C
23 DIMENSION IALTS(12),VS(12)
24 REAL*8 ATTR,CMP,V,VS,VEHIC
25 DATA VEHIC/'VEHICLE'/,VEH/'VEH'/

C
26 CALL NULIFY(ICMPS,NCMPTP)
27 CALL INITVL(VEHS,NVEHS*2,BLS)
28 CALL NULIFY(IRECS,NREC1*NVR)
29 LSTVH = 0
30 LVEH = 0
31 IVEHS = NULL

C
GO THROUGH INTERNAL FILE OF USER SPECIFICATIONS, NOTING THOSE THAT
C REFER TO AN EXISTING VEHICLE AND ARRANGING THEM SO THAT THE REQUIRED
C INFORMATION CAN BE EXTRACTED DURING ONE PASS THROUGH THE EXISTING
C VEHICLE FILE. AT THE SAME TIME, PROCESS ALL SPECIFICATIONS THAT
C DON'T REQUIRE ACCESS TO THE EXISTING VEHICLE FILE.

C
32 IF (LSTCMP .LE. 0) GO TO 50
33 C ITERATE OVER COMPONENT SPECIFICATION RECORDS
34 DO 40 I=1,LSTCMP
35 C CHECK THAT COMPONENT TYPE CODE IS LEGAL
36 IF (TPCMPI(I) .NE. VEH) GO TO 10
37 C THIS RECORD SPECIFIES A BASIC VEHICLE TO START WITH
38 IF (VEHCMP(I) .EQ. BL8) CALL ERR(19,'INVEHC',I,
INPUT ROUTINES - INVEHC SUBROUTINE

C GET INDEX OF VEHICLE IN VEHICLE NAME TABLE
K = ITABLE(VEHS,VEHCMP(I),NVEHS,2)
C FOR EACH COMPONENT OF VEHICLE,
C GET NEW RECORD TO STORE INFORMATION TO GUIDE LATER FILE RETRIEVAL
C & STORE INFORMATION IN THIS RECORD
DO 5 ICMP=1,NCMTP
5 ICMP(S(ICMP)) = NEWVEH(ICMPS(ICMP),0,I,K)
GO TO 40
C
C GET COMPONENT INDEX
ICMP = LSERCH(CMPCOD,TPCMP(I),NCMTPP,I)
C IF ICMP .NE. NULL GO TO 19
CALL ERR(22,'INVEHC',TPCMP(I),I,I)
GO TO 40
C DOES THIS RECORD INDICATE THE NUMBER OF COMPONENTS OF A PARTICULAR
C TYPE DESIRED?
15 IF (NUMCMP(I) .LE. 0) GO TO 20
C IF SO, ADJUST MIN & MAXNUM TO THIS NUMBER
IF (NUMCMP(I) .LT. MINNUM(ICMP) .OR. NUMCMP(I) .GT. MAXNUM(ICMP)) CALL ERR(23,'INVEHC',NUMCMP(I),ICMP,I)
MINNUM(ICMP) = NUMCMP(I)
MAXNUM(ICMP) = NUMCMP(I)
C
C IS AN EXISTING VEHICLE REFERRED TO?
20 IF (VEHCMP(I) .EQ. BL8) GO TO 30
C GET INDEX OF VEHICLE IN VEHICLE NAME TABLE
K = ITABLE(VEHS,VEHCMP(I),NVEHS,2)
C IF SO, GET NEW RECORD TO STORE INFORMATION TO GUIDE LATER FILE
C RETRIEVAL & STORE INFO IN RECORD
ICMP(S(ICMP)) = NEWVEH(ICMPS(ICMP),1,I,K)
GO TO 40
C
C COMPONENT SPEC SHOULD REFER TO AN ALTERNATIVE COMPONENT
INPUT ROUTINES - INVEHC SUBROUTINE

54 30 IF (INDCMP(I) .GE. 1 .AND. INDCMP(I) .LE. NALTCS(ICMP))
C CREATE DUPLICATE RECORD OF THE DESIRED ALTERNATIVE COMPONENT
57 32 IRECTP = ICMP + 1
58 IOFF = IFILPT(RECTP)
59 IR = NEWREC(RECTP)
60 CALL CCPY(VFILE(ICFF + (INDCMP(I)-1)*NFLDS(RECTP)),
C COMPONENTS
61 CALL SUNE(IWECE(ICMP,ISPECV),VFILE(IOFF),NFLDS(RECTP),
C INDICATE COMPONENT TO BE SELECTED
62 IWECE(ICMP,ISPECV) = IR
63 CONTINUE
C IVEHC
50 IF (LSTENG .LE. 0) GO TO 100
C ITERATE OVER ENGINEERING PARAMETER CONSTRAINTS SPECIFIED BY USER
65 DO 90 I=1,LSTENG
C VERIFY THAT COMPONENT TYPE CODE SPECIFIED IS OK & GET CORRESPONDING
C COMPONENT INDEX
66 ICMP = LSERCH(CMPCOD,TYPENG(I),NCMPTP,1)
67 IF (ICMP .NE. NULL) GO TO 55
C VERIFY THAT ATTRIBUTE IS SPECIFIED CORRECTLY; GET CORRESPONDING
C INDEX
70 55 IFLD = IATFL(TRENG(I),ICMP + 1)
71 IF (IFLD .NE. NULL) GO TO 60
C
INPUT ROUTINES - INVEHC SUBROUTINE

C VERIFY THAT RELCP IS APPROPRIATE
74   DO 65 J=1,NRELCP
75   IF (RELENG(I) .EQ. RELNAM(J)) GO TO 70
76   CONTINUE
77   CALL ERR(26,'INVEHC',TYPENG(I),ATRENG(I),RELENG(I))
78   GO TO 90

C 79 KREL = IRELTP(I)

C IS AN EXISTING VEHICLE REFERRED TO ?
80   IF (VEHENG(I) .EQ. BL8) GO TO 80
C GET INDEX OF VEHICLE IN VEHICLE NAME TABLE
81   K = ITABLE(VEHS,VEHENG(I),NVEHS,2)
82   IF (K .EQ. NULL) CALL ERR(2), 'INVEHC',VEHENG(I),NVEHS,0)
83   IF (K .GT. LVEH) LVEH = K

C IF SO, GET NEW RECORD AND STORE INFO IN IT TO GUIDE LATER FILE
C RETRIEVAL
84   ICMPS(ICMP) = NEWVEH(ICMPS(ICMP),2,I,K)
85   GO TO 90

C ENGINEERING PARAMETER SPEC SHOULD SPECIFY A VALUE. STORE THIS
C AS A CONSTRAINT
86   IR = NEWREL(ICMP+1,IFLD,KREL,VL1ENG(I),INDENG(I))
87   90   CONTINUE
C
C PROCESS PERFORMANCE PARAMETER SPECIFICATIONS
88   100 IF (LSTPER .LE. 0) GO TO 150
C ITERATE OVER PERFORMANCE CONSTRAINTS SPECIFIED BY USER
89   DO 140 I=1,LSTPER
C
C VERIFY THAT ATTRIBUTE IS SPECIFIED CORRECTLY; GET CORRESPONDING INDEX
90   IFLD = IATTR2(ATRPER(I),1)
91   IF (IFLD .NE. NULL) GO TO 110
92   CALL ERR(25,'INVEHC',VEH,ATRPER(I),0)
93   GO TO 140
INPUT ROUTINES - INVEHC SUBROUTINE

C VERIFY THAT RELCP IS APPROPRIATE
94 110 DO 115 J=1,NRELOP
95 115 IF (RELPER(I) .EQ. RELNAM(J)) GO TO 120
96 115 CONTINUE
97 115 CALL EPR(26,'INVEHC',VEH,ATPER(I),RELPER(I))
98 115 GO TO 140
C
99 120 KREL = IRELTID(J)
C IS AN EXISTING VEHICLE REFERRED TO?
100 130 IF (VEHPER(I) .EQ. BL8) GO TO 130
101 130 K = ITABLE(VEHS,VEHPER(I),NVEHS,2)
102 130 IF (K .EQ. NULL) CALL ERR(2,'INVEHC',VEHPER(I),NVEHS,0)
C IF SO, GET NEW RECORD AND STORE INFO IN IT TO GUIDE LATER FILE
C RETRIEVAL
103 130 IVEHS = NEWVEH(IVEHS,3,I,K)
104 130 GO TO 140
C PERFORMANCE PARAMETER SPEC SHOULD SPECIFY A VALUE. STORE THIS AS
C A CONSTRAINT.
105 140 IR = NEWREL(1,IFLD,KREL,V1PER(I),0)
106 140 CONTINUE
C
C READ & EXTRACT INFORMATION FROM EXISTING VEHICLE FILE
C
107 150 IF (VEHS(1) .EQ. BL8) GO TO 500
C
READ VEHICLE CATEGORY HEADER AND CHECK AGAINST EXPECTATION
C
109 500 READ (2,900,END=500) V
109 900 FORMAT (A8)
110 900 IF (V NE. VEHCAT) CALL ERR(18,'INVEHC',V,VEHCAT,0)
INPUT ROUTINES - INVEHC SUBROUTINE

C DETERMINE THE LAST COMPONENT THAT NEED TO READ INFORMATION ABOUT
C
111  LCMP = 0
112  DO 155 ICMP=1,NCMPTP
113       IF (ICMPS(ICMP) .NE. NULL) LCMP = ICMP
114  155  CONTINUE
C
C READ VEHICLE RECORD
115  READ (2,910,END=380) V,VS
116  910  FORMAT (A8,4X,12(A8,2X))
C
C IDENTIFY THE VEHICLES WHICH WANT TO EXTRACT INFORMATION ABOUT
117  DO 180 IV=1,LVEH
118       DO 165 ICCL=1,NCPCL
119          IF (VS(ICOL) .EQ. VEHS(IV)) GO TO 170
120       165  CONTINUE
121       GO TO 180
C
C FOUND A VEHICLE OF INTEREST; MARK IT FOUND & REMEMBER ITS COLUMN
122  170   ICOLS(IV) = ICOL
123  180  CONTINUE
C
C READ VEHICLE COMPONENT POINTERS
124  190  READ (2,920) CMP,IVALTS
125  920  FORMAT (2X,A8,12(A8,12))
126       IF (CMP .EQ. BL8 .OR. CMP .EQ. DASHES) GO TO 220
C
C IDENTIFY WHETHER THIS IS A COMPONENT OF INTEREST
127  DO 200 ICMP=1,NCMPTP
128       IF (ICMPS(ICMP) .NE. NULL .AND. RECNAM(ICMP+1) .EQ. CMP)
129          +            GO TO 205
129  200  CONTINUE
130       GO TO 190
C
C COMPONENT IS OF INTEREST; REMEMBER WHICH COLUMN
C EACH COMPONENT OF INTEREST OF THIS TYPE IS IN.
INPUT ROUTINES — INVEHC SUBROUTINE

C ITERATE OVER CONSTRAINTS INVOLVING THIS COMPONENT & STORE COLUMN
C INFORMATION

131 205  IREC = ICMP$1(ICMP)
132 208  IF (IREC .EQ. NULL) GO TO 190
133 190  IV = IRECS(JVEH,IREC)
134 191  ICOL = ICOLS(IV)
135 192  IF (IRECS(JCOL,IREC) .EQ. NULL) IRECS(JCOL,IREC) = IALTS(ICOL)
136 193  IREC = IRECS(JNXT,IREC)
137 208  GO TO 208

C
C READ VEHICLE PERFORMANCE PARAMETERS

138 221  READ (2,93) ATTR
139 222  930  FORMAT (2X,A8)
140 223  IF (ATTR .EQ. BL8 .OR. ATTR .EQ. DASHES) GO TO 240

C IS THIS ATTRIBUTE OF INTEREST?

141 225  IREC = IVEHS
142 226  225  IF (IREC .EQ. NULL) GO TO 226
143 227  ISPEC = IRECS(JSPEC,IREC)
144 228  IF (ATTRPER(ISPEC) .NE. ATTR) GO TO 230

C THIS IS ATTRIBUTE OF INTEREST; READ IT

145 230  CALL INATTR(ATTR,IREC,ISPEC,1)
146 231  230  IREC = IRECS(JNXT,IREC)
147 232  GO TO 225

C
C READ VEHICLE COMPONENT HEADER

148 240  READ (2,947,END=300) CMP,IALTS
149 940  FORMAT (A8,2X,12(8X,12))

C DETERMINE COMPONENT TYPE

150 250  DO 250 ICMP=1,NCMTP
151 251  IF (RECNAM(ICMP+1) .EQ. CMP) GO TO 255
152 252  CONTINUE
153 255  CALL ERR(28,'INVEHC',CMP,0,0)

C
C IS THIS A COMPONENT OF INTEREST?

154 255  IF (ICMP .GT. LCMP) GO TO 300
155 300  IF (ICMPS(ICMP) .NE. NULL) GO TO 260
INPUT ROUTINES - INVEHC SUBROUTINE

C
C IF NOT, SKIP OVER ITS ATTRIBUTE RECORDS
156  258  READ (2,930,END=300) ATTR
157     IF (ATTR .EQ. BL8 .OR. ATTR .EQ. DASHES) GO TO 240
158     GO TO 258
C
C ARE INTERESTED IN THIS COMPONENT. ITERATE OVER USER
C SPECIFICATIONS FOR EACH ATTRIBUTE OF THE COMPONENT TO SEE WHETHER
C THE ATTRIBUTE IS OF INTEREST. FIRST READ THE NEXT ATTRIBUTE.
C
159  260  IRECTP = ICMP + 1
160  262  READ (2,930,END=300) ATTR
161     IF (ATTR .EQ. BL8 .OR. ATTR .EQ. DASHES) GO TO 240
C
C IS THIS AN ATTRIBUTE OF INTEREST?
162  265  IREC = ICMP$ICMP
163  265  IF (IREC .EQ. NULL) GO TO 262
164  265  ISPEC = IRECS(IJSPEC,IREC)
165  265  ITYP = IRECS(ITYP,IREC)
166  265  IF (ITYP .EQ. 2) GC TO 270
C
C SPECIFICATION CALLS FOR THIS ATTRIBUTE TO BE USED AND STORED
C AS A CONSTANT
167  265  CALL IINATR2(ATTR,IREC,ISPEC,IRECTP)
168  265  GO TO 275
C
C CHECK WHETHER ATTRIBUTE IS OF INTEREST
169  270  IF (ATTR .NE. ATRENG(ISPEC)) GO TO 275
C
C SPECIFICATION CALLS FOR THIS ATTRIBUTE TO BE READ AND STORED AS
C A CONSTRAINT
170  275  CALL INATR1(ATTR,IREC,ISPEC,IRECTP)
C
C GET NEXT SPECIFICATION RELEVANT TO THIS COMPONENT
171  275  IREC = IRECS(JNXT,IREC)
172  275  GO TO 265

C
C CHECK THAT ALL USER SPECIFICATIONS REFERING TO THE
C EXISTING VEHICLE FILE HAVE BEEN PROCESSED.
173  300  IPREV = NULL
INPUT ROUTINES - INVEHC SUBROUTINE

ISN
174 DO 310 IV=1,LSTVEH
175 IF (IRECS(JDONE,IV),NE. NULL) GO TO 310
176 IRECS(JDONE,IV) = IPREV
177 IPREV = IV
178 310 CONTINUE
179 IF (IPREV .NE. NULL) CALL ERR(32,'INVEHC',IPREV,0,0)
180 RETURN
C
181 380 CALL ERR(33,'INVEHC',LSTVEH,0,0)
C
182 500 RETURN
183 END
C.4: SOLUTION ROUTINES

This section contains listings of the solution routines including the main "backtracking" algorithm (GENVEH), the function OKCOMP used to check compatibility of components with user specifications, the function COMPAT, used to check component compatibility and the routines used to estimate vehicle engineering and performance parameters (DIMENS, FPOWER, MOBILE, and PAIT). See section C-2 for listings of the labeled COMMON blocks referred to in these routines.
SOLUTION ROUTINES - FUNCTION COMPAT

ISN
1
LOGICAL FUNCTION COMPAT(ICMP, IR)

C
ROUTINE TO TEST WHETHER A TRIAL COMPONENT OF A GIVEN TYPE IS
COMPATIBLE WITH PREVIOUS COMPONENTS SELECTED. RETURNS .TRUE. IF SO,
.FALSE. OTHERWISE.

C
INPUT PARAMETERS:
C
ICMP COMPONENT TYPE
C
IR COMPONENT

C
COMMON /VEH/
3
DIMENSION IVECLE(72, 3), IHULL(48, 9), ITURRET(42, 9), FMAING(44, 9),
+ FMACGN(40, 10), FMISGN(40, 9), IADGUN(40, 6), IRANGR(28, 7),
+ ISENS(32, 9), ISTABEL(30, 6), IIGNCT(28, 9), IAMMO(30, 30),
+ IAMMIC(26, 9), IENG(40, 8), ITOPS(32, 8), IFINDR(28, 10),
+ IROADW(36, 10), ISPRNG(30, 10), ITRACK(34, 8), ISKIRT(28, 8),
+ IFUEL(28, 10), IPFUEL(26, 9), ICREW(26, 5), ICARGO(26, 5),
+ TECR(26, 5), ICNTH(28, 3), IFIREX(28, 13), IENVIR(26, 9),
+ IDIAGN(26, 5), ISIGSP(26, 24), ITSMAKE(26, 10), TEWSYS(26, 10),
+ THL(3, 5), VFILR(1), IVFILR(1)

4
EQUIVALENCE (VECLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
+ ITURRET(1)), (MAING(1), FMAING(1)), (MAGUN(1), FMACGN(1)),
+ (MISGN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
+ IRANGR(1)), (SENS(1), ISENS(1)), (STABLE(1), ISTABEL(1)),
+ (IGNCT(1), IAMMO(1)), (AMMO(1), IAMMO(1)), (AMMIC(1),
+ IAMMIC(1)), (ENG(1), IENG(1)), (TRANS(1), ITRANS(1)),
+ (PFLIND(1), IFINDR(1)), (ROAH(1), IROADW(1)), (SPRING(1),
+ ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),
+ (FUEL(1), IFUEL(1)), (FUEL(1), IPFUEL(1)), (CREW(1),
+ ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
+ (COM(1), ICNTH(1)), (IFIREX(1), IFIREX(1)), (ENVIRC(1),
+ IENVIR(1)), (DIAGN(1), IDIAGN(1)), (SIGSP(1), ISIGSP(1)),
+ (SMAKE(1), ITSMAKE(1)), (TEWSYS(1), TEWSYS(1)), (RELN(1), IRELN(1))

5
EQUIVALENCE (VECLE(1), VFILR(1), IVFILR(1))

6
COMMON /VPROTO/
SOLUTION ROUTINES - FUNCTION COMPAT

TSN
7 REAL*8 PLDNAME, RECHAM, DEFAULT
8 COMMON /JPARS/
9 COMMON /JPARN/
10 COMMON /NPARS /
11 COMMON /AUX /
12 REAL*8 BL8, DASHES
13 EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
14 COMMON /SPECS /
15 REAL*8 ATRENG, ATKPER, VEHCAT, VEHCMP, VEHENG, VEHPER

C
16 COMPAT = .TRUE.
17 GO TO (90, 90, 10, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90)

C
18 1) WMIN = FMAING(JMINWT, IR)
19 IF (WMIN .EQ. FNULL) GO TO 90
20 IF (WMIN .GT. VEHWT) COMPAT = .FALSE.
21 GO TO 90

C
18 CHECK THAT MAIN GUN IS NOT TOO LARGE FOR WEIGHT OF VEHICLE
C
19 CHECK THAT ORDUNCE HAS SAME CALIBER AS AT LEAST ONE OF THE GUNS
C
20 CAL = DVAL2 (AMMO(JCALIB, IR), ICMP, JCALIB, 'MM '
21 DO 26 IC = JMAIN, JADGUN
22 IGR = IPvCLE(IC, 1CANDV)
23 IF (IGR .EQ. NULL) GO TO 26
24 TRECTP = IC + 1
25 IOPF = IFILTP(TRECTP)
26 GCAL = DVAL(VFILE(IOPF),NFLDS(TRECTP), JCALIB, IGR, 'MM '
27 IF (GCAL - CAL .LT. .001 AND. CAL-GCAL .LT. .001) GO TO 90
28 IGR = IDVAL(VFILE(IOPF), NFLDS(TRECTP), JNEXT, IGR, 'MM '
29 GO TO 22
30 CONTINUE
31 COMPAT = .FALSE.
32 GO TO 90
33 C
34 GO TO 90
SOLUTION ROUTINES - FUNCTION COMPAT

C CHECK THAT TRANSMISSION IS COMPATIBLE WITH ENGINE
35 30 KEYE = IDVAL (ENGINE, NFLDS (JENGIN+1), JTRNRQ, IVECLE (JENGIN, ICANDV),
   + JENGIN+1,'-',DF)
36 KEYT = TDVAL2 (TRANSH (JKEY, IR), JTRANS, JKEY, ' ')
37 IF (KEYE .NE. KEYT) COMPAT = .FALSE.
38 GO TO 91
C
39 91 RETURN
40 END
SOLUTION ROUTINES - DIMENS SUBROUTINE

SUBROUTINE DIMENS (IV, IRETH)

ROUTINE TO CALCULATE THE DIMENSIONS OF A TANK BASED ON AN ASSUMED
GEOMETRICAL CONFIGURATION (SEE ARTICLE BY R. LAWSON IN THE TANK
WEAPON SYSTEM, SYSTEM RESEARCH GROUP, OSU, SEPT. 1968)
ALSO CALCULATES SELECTED MOBILITY CHARACTERISTICS.

INPUT PARAMETERS:
IV SPECIFIC VEHICLE OF INTEREST

OUTPUT PARAMETERS:
IRETH COMPONENT TYPE TO TRY NEW SELECTION IF BACKUP IS NEEDED;
SET TO ZERO OTHERWISE.

COMMON /VEH/
DIMENSION IVVECLE (72, 3), IHULL (48, 9), ITURRET (42, 9), FMAING (44, 9),
+ FMAGN (40, 10), FADGUN (40, 7), IADGUN (40, 6), IRANGR (28, 7),
+ TSENSR (32, 9), ISTALL (30, 6), IGNCTL (28, 9), IAMMC (30, 30),
+ IAMMC (26, 9), IENGIN (40, 8), ITRANS (32, 8), IFINDR (28, 10),
+ IROADW (36, 10), ISPRNG (30, 10), ITRACK (34, 8), ISKITR (28, 8),
+ IFUEL (28, 10), IFUELC (26, 9), ICREW (26, 5), ICARGO (26, 5),
+ ISELECT (26, 6), ICOMMO (28, 9), IFIREX (28, 10), IENVIR (26, 9),
+ IIDIAGN (26, 5), IISIGSP (26, 24), IMOKE (26, 10), IIEWSYS (26, 10),
+ IRELN (3, 50), VFILE (1), VPPLE (1)

EQUIVALENCE (VEHICLE (1), IVVECLE (1)), (HULL (1), IHULL (1)), (TURRET (1),
+ ITURRET (1)), (MAING (1), FMAING (1)), (MACGUN (1), FMACGN (1),
+ (MISSN (1), FALSN (1)), (ADGUN (1), IADGUN (1)), (RANGER (1),
+ IRANGR (1)), (SENSE (1), ISENSE (1)), (STABLE (1), ISTALL (1),
+ (GUNCTL (1), IGNCTL (1)), (AMMO (1), IAMMC (1)), (AMMC (1),
+ IAMMC (1)), (ENGINE (1), IENGIN (1)), (TRANSM (1), ITRANS (1),
+ (FINDR (1), IFINDR (1)), (ROADW (1), IROADW (1)), (SPRING (1),
+ ISPRNG (1)), (TRACK (1), ITRACK (1)), (SKIRT (1), ISKITR (1),
+ (FUEL (1), IFUEL (1)), (FUELC (1), IFUELC (1)), (CREW (1),
+ ICREW (1)), (CARGO (1), ICARGO (1)), (ELECTR (1), ISELECT (1),
+ (COMO (1), ICOMMO (1)), (FIREX (1), IFIREX (1)), (ENVIR (1),
+ (RELN (1), VFILE (1), VPPLE (1))
SOLUTION ROUTINES - DIMENS SUBROUTINE

C
+ TENVTR(1), (DIAGNO(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
+ (SHEME(1)), (EWSys(1)), (EWsys(1)), (REL(1), IREL(1))
5 EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))
6 COMMON /VPROTO/
7 REAL*3 PLOAD, BECHAM, DEFAULT
8 COMMON /JPARS/
9 COMMON /JPARS1/
10 COMMON /NPARS/
11 COMMON /AUX/
12 REAL*3 BLA, DASHES
13 EQUIVALENCE (FNULL, NULL), (IBLS, BL5)
14 COMMON /ARRAYS/
15 REAL*3 ANDF, MODEL, NATION
16 DIMENSION ARRAY(1)
17 EQUIVALENCE (ABRAY(1), DETRNG(1))
18 COMMON /JPARS2/
19 COMMON /DATA/
20 COMMON /DATA2/

C
DATA PI = 3.14159/
C
C GET POINTERS TO COMPONENTS OF VEHICLE
21 LTURF = IVECLE(JTURF, IV)
22 LHULL = IVECLE(JHULL, IV)
23 LMAING = IVECLE(JMAING, IV)
24 LRENG = IVECLE(JRENG, IV)
25 LTRANS = IVECLE(JTRANS, IV)
26 LPINDR = IVECLE(JPINDR, IV)
27 LROADM = IVECLE(JROADM, IV)
28 LTRACK = IVECLE(JTRACK, IV)
29 LSKIRT = IVECLE(JSKIRT, IV)
30 LAMM = IVECLE(JAMM, IV)
C
C TURRET DIAMETER
31 X1 = DVAL2('TALRN(JX1, LHAING), JMAING, JX1, 'IN ')
33 X2 = DVAL2('TURRET(JX2, LTURF), JTURF, JX2, 'IN ')
C
SOLUTION ROUTINES - DIMENS SUBROUTINE

C

34  XTP = 2. * (X1+X2)
35  TURRET(JXT,P,LTURET) = XTP

C

C HULL INSIDE WIDTH & VEHICLE WIDTH
36  X5 = DVAL2(ENGINE(JX5,LENGIN),JENGIN,JX5,'IN')
37  X6 = DVAL2(ENGINE(JX6,LENGIN),JENGIN,JX6,'IN')
38  X7 = DVAL2(TRASM(JX7,LTRANS),JTRANS,JX7,'IN')
39  X8 = DVAL2(FINDR(JX8,LFINDR),JFINDR,JX8,'IN')
40  XHPT = AMAX1(X5+2.*X6,X7+2.*X8)
41  XH = AMAX1(TURRET(JXT,P,LTURET),XHPT)
42  HULL(JXH,LHULL) = XH
43  XJ = DVAL2(HULL(JX3,LHULL),JHULL,JX3,'IN')
44  X4 = DVAL2(TRACK(JX4,LTRACK),JTRACK,JX4,'IN')
45  X7 = 2.*X3 + 2.*X4 + XH
46  VEHICLE(JWVND,IV) = X0

C

C LENGTH OF HULL
47  JTHFU = DVAL2(HULL(JTHFU,LHULL),JHULL,JTHFU,'IN')
48  JTHFL = DVAL2(HULL(JTHFL,LHULL),JHULL,JTHFL,'IN')
49  GAMMAU = DVAL2(HULL(JGAMU,LHULL),JHULL,JGAMU,'RAD')
50  COSGU = COS(GAMMAU)
51  TANGU = TAN(GAMMAU)
52  Y1 = JTHPU / COSGU
53  Y2 = DVAL2(HULL(JX2,LHULL),JHULL,JY2,'IN')
54  Y3 = DVAL2(HULL(JX3,LHULL),JHULL,JY3,'IN')
55  K1 = DVAL2(TURRET(JK1,LTURET),JTURET,JK1,'IN')
56  XTP = DVAL2(TURRET(JXT,P,LTURET),JTURET,JXTP,'IN')
57  Y4 = XTP + K1
58  Y5 = DVAL2(HULL(JY5,LHULL),JHULL,JY5,'IN')
59  Y6 = DVAL2(ENGINE(JY6,LENGIN),JENGIN,JY6,'IN')
60  Y7 = DVAL2(TRASM(JY7,LTRANS),JTRANS,JY7,'IN')
61  Y8 = DVAL2(TRASM(JY8,LTRANS),JTRANS,JY8,'IN')
62  Y9 = DVAL2(HULL(JY9,LHULL),JHULL,JY9,'IN')
63  YH = Y2 + Y3 + Y4 + Y5 + Y6 + Y7 + Y8
64  HULL(JYH,LHULL) = YH
65  Y0 = Y1 + YH + Y9
SOLUTION ROUTINES - DIMENS SUBROUTINE

VENVLE(JVLN,IV) = YJ

C TRACK GROUND CONTACT LENGTH

D1 = DVAL2(ROADW1(JD1,LRoadw),JRводw,JD1,'IN',
D2 = DVAL2(ROADW1(JD2,LRoadw),JRводw,JD2,'IN',
D3 = DVAL2(ROADW1(JD3,LRoadw),JRводw,JD3,'IN',
T = DVAL2(TRACK(JT,LRack),JTrack,JT,'IN',
Z1 = DVAL2(ROADW1(JZ1,LRoadw),JRводw,JZ1,'IN',
Z14 = DVAL2(ROADW1(JZ14,LRoadw),JRoadw,JZ14,'IN',
Z15 = DVAL2(ROADW1(JZ15,LRoadw),JRoadw,JZ15,'IN',
BF = DVAL2(ROADW1(JBetaF,LRoadw),JRoadw,JBetaF,'RAD',
BR = DVAL2(ROADW1(JBetaR,LRoadw),JRoadw,JBetaR,'RAD',
SINBF = SIN(BF)
TANBF = TAN(BF)
COSBF = SINBF / TANBF
SINBR = SIN(BR)
TANBR = TAN(BR)
COSBR = SINBR / TANBR
TD3Z1 = T + D3/2. - Z1
ZZ14 = SQRT((D1-D3)/2.)*2 - ((D1-D3)/2.)*2
+ (D1-D3) / (2.*TANBF) * SINBF * TD3Z1
ZZ15 = SQRT((D2-D3)/2.)*2 - ((D2-D3)/2.)*2
+ (D2-D3) / (2.*TANBR) * SINBR * TD3Z1

IF (Z14 .GE. ZZ14) GO TO 5
CALL ERR(40,'DIMENS','Z14','IDLER',Z14)
Z14 = ZZ14

5 IF (Z15 .GE. ZZ15) GO TO 10
CALL ERR(40,'DIMENS','Z15','SPOCKET',Z15)
Z15 = ZZ15

1) C1 = D1/2. + T
C2 = D2/2. + T
Z1M = Z1 - (T + D3/2.)
YGC = Y0 - (C1 + C2 +
+ (2.*COSBF*(Z14+Z1M) - (D1-D3)) / (2.*SINBF) +
+ (2.*COSBR*(Z15+Z1M) - (D1-D3)) / (2.*SINBR))

VEHICLE(JYGC,IV) = YGC / 12.
SOLUTION ROUTINES - DIMENS SUBROUTINE

C

C HEIGHTS

96  Z2 = DVAL2 (HULL (JZ2, LHULL), JHULL, JZ2, 'IN ')
97  Z3 = DVAL2 (HULL (JZ3, LHULL), JHULL, JZ3, 'IN ')
98  Z4 = DVAL2 (HULL (JZ4, LHULL), JHULL, JZ4, 'IN ')
99  Z5 = DVAL2 (TURRET (JZ5, LTURRET), JTURRET, JZ5, 'IN ')
100 Z10 = DVAL2 (ENGINE (JZ10, LENGINE), JENGINE, JZ10, 'IN ')
101 Z11 = DVAL2 (ENGINE (JZ11, LENGINE), JENGINE, JZ11, 'IN ')
102 Z12 = DVAL2 (HULL (JZ12, LHULL), JHULL, JZ12, 'IN ')
103 Z13 = DVAL2 (TURRET (JZ13, LTURRET), JTURRET, JZ13, 'IN ')

C

ALPHAP = DVAL2 (MAINGN (JDPRES, DMAING), DMAING, JDPRES, 'RAD ')
106  B = DVAL2 (MAINGN (JODIAN, DMAING), DMAING, JODIAN, 'IN ')
107  Y22 = Y1 + Y2 + Y3 + Y4/2. - (Z1+Z2+Z4)*TANGU/2.
108  Y23 = DVAL2 (TURRET (JY23, LTURRET), JTURRET, JY23, 'IN ')
109  Y24 = Y4/2. + Y5 + Y6 + Y7 + Y8
110  SINA = SIN (ALPHAP)
111  B2SAF = B/(2. * SINA)
112  TANA = TAN (ALPHAP)
113  COSET = SINA / TANA
114  Z32 = (X0/2.) ** 2
115  Zz5 = (SQR (Y22**2 + X02) + B2SAF - Y23)*TANA
118  CALL WARN (2, 'DIMENS', 'FRONT', ZZ5, Z5)
119    Z5 = ZZ5
120  15  Z213 = (SQR (Y24**2 + X02) + B2SAF - Y23)*TANA
121  IF (TURRET (JZ13, LTURRET) .EQ. FNNULL) GO TO 18
122  IF (Z13 .GE. Z213) GO TO 20
123  CALL WARN (2, 'DIMENS', 'REAR', ZZ13, Z13)
124  18  Z13 = ZZ13
125  20  TURRET (JZ13, LTURRET) = Z13
126  21  Y20 = DVAL2 (MAINGN (JY20, DMAING), DMAING, JY20, 'IN ')
127  Y21 = DVAL2 (MAINGN (JY21, DMAING), DMAING, JY21, 'IN ')
128  Z6 = (Y20+Y21)*SINA + B/(2.*COSET)
SOLUTION ROUTINES - DIMENS SUBROUTINE

C

ISN

129  Z7 = DVAL2(TURRET(JZ7,LTURRET),JLTURRET,JZ7,'IN ')         
130  Z8 = DVAL2(HULL(JZ8,LHULL),JHULL,JZ8,'IN ')              
131  Z9 = DVAL2(HULL(JZ9,LHULL),JHULL,JZ9,'IN ')              
132  ZMAX2 = AMAX1(Z3+Z4+Z5+Z6,Z10+Z11+Z12+Z13+Z6,Z8+Z9)   
133  Z0 = Z1 + Z2 + ZMAX2 + Z7                               
134  VEHICLE(JVHT,IV) = Z0                                    

C

AREA & WEIGHT OF HULL

135  ZH = AMAX1(Z3,Z10+Z11)                                    
136  HULL(JZM,LHULL) = ZH                                       
137  TARMHL = TNULL(JAKMTP,LHULL)                              
138  D = DENSEY(TARMHL)                                        
139  AHUF = XH*ZH/(2. * COSGU)                                 
140  WHUF = AHUF * TPHU * D                                     

C

142  WHFD = AHFD * Z4 * D                                       

C

144  WHRD = AHRD * Z12 * D                                      

C

145  AHUB = XH*ZH / 2.                                          
146  WHUB = AHUB * Y9 * D                                       

C

147  DELTAD = DVAL2(HULL(JDELD,LHULL),JHULL,JDELD,'RAD ')      
148  COSDD = COS(DELTAD)                                       
149  TANDD = TAN(DELTAD)                                       
150  AHB = XH*ZH/(2. * COSDD)                                  
151  WHB = AHB * Y9 * D                                        

C

152  GAMMAD = DVAL2(HULL(JGAMD,LHULL),JHULL,JGAMD,'RAD ')      
153  TANGD = TAN(GAMMAD)                                       
154  COSGD = COS(GAMMAD)                                       
155  AHB = XH*(YH - ZH*TANDD/2. - ZU*TANGD/2.)                 
156  WHB = AHB * Z2 * D                                        

C
SOLUTION ROUTINES - DMENS SUBROUTINE

C
157 AHS = 2. * YH * ZH - (ZH**2/2.) * (TANDD + TANGD + TANGU)
158 WHS = AHS * X3 * D
C
159 AHLF = YH * ZH / (2. * COSGD)
160 WHLF = AHLF * THFL * D
C
161 WH = WHUP + WHFD + WHRD + WHUB + WHLB + WHR + WHS + WHLF
162 HULL (JWT, LHULL) = WH
C
C WEIGHT OF TURRET ARMOR

163 AT = .717 * Y4
164 TURRET (JAI, LTURET) = 2. * AI
165 CI = .59 * Y4
166 BI = 20 - (Z1 + Z2 + Z3 + Z4 + Z7)
167 TURRET (JDI, LTURET) = 2. * BI
168 IARMT = IDVAL2 (TURRET (JARMT, LTURET), JTURET, JARMT, 'IN '
169 DT = DENSY (IARMT)
170 TTF = DVAL2 (TURRET (JTTT, LTURET), JTURET, JTTT, 'IN '
171 TTS = DVAL2 (TURRET (JTTT, LTURET), JTURET, JTTT, 'IN '
172 TTD = DVAL2 (TURRET (JTTD, LTURET), JTURET, JTTD, 'IN '
173 TTB = DVAL2 (TURRET (JTTB, LTURET), JTURET, JTTB, 'IN '
174 WT = .6667 * PI * ((AI + TTF/2. + TTB/2.) * (BI + Z7) * (CI + TTS)
175 + (I*AI*CI) * DT + (4.*PI*AI*CI - PI*Y4**2/4.) * TTD*DT
176 TURRET (JWT, LTURET) = WT
C
C COMPUTE WEIGHT OF TRACK SKIRTS, IF ANY

176 IF (LSKIRT .EQ. NULL) GO TO 30
177 IARMT = IDVAL2 (SKIRT (JTYPE, LSKIRT), JSKIRT, JTYPE, 'IN '
178 DS = DENSY (IARMT)
179 HS = DVAL2 (SKIRT (JSKHT, LSKIRT), JSKIRT, JSKHT, 'IN '
180 TS = DVAL2 (SKIRT (JSKTHK, LSKIRT), JSKIRT, JSKTHK, 'IN '
181 WS = 2.*DS*TS*(Y0*HS + Y0*X4 + 2.*X4*HS)
182 SKIRT (JWT, LSKIRT) = WS
SOLUTION ROUTINES - DIMENS SUBROUTINE

C SUM WEIGHT OF ALL COMPONENTS
183  3)  W = SUMCMC(IP,JWT,'LB ')
184      WT0N = W / 2000.
185      VEHCL(E,JWIVT,IV) = W
C
C SUM INTERNAL VOLUMES OF COMPONENTS
186      VSUM = SUMCMC(IP,JVOL,'FT3 ')
C
C CALCULATE INTERNAL VOLUME OF THE VEHICLE ENVELOPE
187      VH = AHS * XH / 2.
188      HUII(JVOL,LHULL) = VH / 1728.
189      VT = .6667 * PI * AI * BI * CI
190      TURRET(JVOL,LTURRET) = VT / 1728.
191      VOLENV = VH + VT
192      VOLPT3 = VOLENV / 1728.
193      VEHCL(JVWOL,IV) = VOLPT3
C
C MOBILITY PARAMETERS
C
C TREAD & L/T RATIO
194      TREAD = VEHCL(JVWID,IV) - X4
195      RLT = YGC / TREAD
196      VEHCL(JLT,IV) = RLT
C
C GROSS HP / TON
197      HP = DVAL2(ENGINE(JHP,LENGIN),LENGIN,JHP,'HP ')
198      HPTCN = HP / WTON
199      VEHCL(JGHPTN,IV) = HPTCN
C
C SPROCKET HP / TON
200      EPFR = DVAL2(TRANS(JEFFIC,LTRANS),LTRANS,JEFFIC,'-'  ')
201      EPPFD = DVAL2(FINLDR(JEFFIC,LFINDR),JFINDR,JEFFIC,'-'  ')
202      SHTPTN = HP * (EPFR/100.) * (EPPFD/100.) / WTON
203      VEHCL(JSHPTN,IV) = SHTPTN
C
SOLUTION ROUTINES - DIMENS SUBROUTINE

C AVERAGE GROUND PRESSURE
204  P = W / (2.*Y4*YGC)
205  VEHICLE(JGPRES,IV) = P
C
C WT PER LINEAL FOOT
206  WPT = WTON / (YGC/12.)
207  VEHICLE(JTONPT,IV) = WPT
C
C MAX DITCH THAT CAN CROSS
208  DITCH = (Y0 - (D1/2.) - (D2/2.) - 2.*T) / 24.
209  VEHICLE(JDWDIV,IV) = DITCH
C
C MAX SPEED ON LEVEL ROAD
210  SPD1 = FVAL1(SPEED(1,1),RSHPTN,NSHPTN,SHPTON)
211  VEHICLE(J4XSPD,IV) = SPD1
C
C MAX SPEED ON 30 DEG GRADE (NO SLIPPAGE)
212  SPD2 = FVAL1(SPEED(1,2),RSHPTN,NSHPTN,SHPTON)
213  VEHICLE(J30SPD,IV) = SPD2
C
C MAX SLOPE THAT CAN CLIMB
214  SLOPMX = FVAL2(SLOPE,RSHPTN,NSHPTN,SHPTON,RVEHWT,NVEHWT,WTON)
215  VEHICLE(JSLOPE,IV) = SLOPMX
C
C ACCELERATION SEC FROM 0-20MPH ON LEVEL ROAD
216  ACC = FVAL2(ACCEL,RSHPTN,NSHPTN,SHPTON,RVEHWT,NVEHWT,WTON)
217  VEHICLE(JACCEL,IV) = ACC
C
C RANGE. FIRST DETERMINE THE TOTAL VOLUME OF FUEL TANKS & THE TOTAL
C VOLUME OF PROTECTED FUEL TANKS
218  IREC = IVehicle(JFUELC,IV)
219  TFUEL = 0.
220  FUELP = 0.
221     IF (IREC .EQ. NULL) GO TO 50
222        GAL = DVAL2(FUELC(JFUEC,IREC),JFUELC,JFUEC,'GAL ')
223        LOC = IDVAL2(IFUELC(JLOC,IREC),JFUELC,JLOC,'- ')

C-77
SOLUTION ROUTINES - DIMENS SUBROUTINE

ISN

224 IF (LOC .LE. 2) FUELP = FUELP + GAL
225 TFUEL = TFUEL + GAL
226 TREC = TREC (JNEXT, IREC)
227 GO TO 45
228 IENGT = IDVAL2 (IENGINE, JTYPE, LENGINE, JENGINE, JTYPE, 'M', 'M')
229 XNMPG = FVAL2 (XMPG (1, 1, IENGT), MPH, NSHPN, SHPNTN, NHTPON, + RVEHWT, NVEHWT, WHTN)
230 RANGE = XNMPG * TFUEL
231 RANGEP = XNMPG * FUELP
232 VEHICLE (JRAF, IV) = RANGE
233 VEHICLE (JRAF, IV) = RANGEP

C ROUGH TERRAIN VELOCITY
234 NUMWHL = IDVAL2 (JROADW, JNERYL, JROADW, JNERYL, 'M', 'M')
235 LSPDMP = IVECLE (JSUSP, IV)
236 WHTRV = FVAL2 (SPRING, JWHTRV, LSPDMP, JSUSP, JWHTRV, 'IN')
237 IF (NUMWHL .LT. MINWHL .OR. NUMWHL .GT. MAXWHL) CALL ERR (41, + 'DIMENS', NUMWHL, 0, 0)
238 NW = NUMWHL - MINWHL + 1
239 VRIDEB1 = FVAL3 (VALID, 1, 1, 1, NW, 1, RWHTRV, NWHTTRV, WHTRV, + RVEHWT, NVEHWT, WTON, RTKLEN, NTKLEN, YGC)
240 VRIDEB2 = FVAL3 (VALID, 1, 1, 1, NW, 2, RWHTRV, NWHTTRV, WHTRV, + RVEHWT, NVEHWT, WTON, RTKLEN, NTKLEN, YGC)

C PROBABILITY OF VEHICLE BEING PENETRATED BY A HIT

241 CALL IVITVL (IPENET, 1, 1, 1, IV, 18, PNULL)
242 DO 60 IORD = 1, NONDT2
243 DO 60 TASP = 1, TASC
244 DO 60 IRANG = 1, NRANG
245 PPHULL = FVAL1 (PHULL, 1, IRANG, TASP, IORD, TARMH), RARMTH, + NARMTH, TYP)  
246 PPTUR = FVAL1 (PPTUR, 1, IRANG, TASP, IORD, TARMT), RARMTH, + NARMTH, TYP) 
247 PIPENET (IRANG, TASP, IORD, IV) = PPHULL * PPHULL + (1. - PPHULL) * + PPTUR
SOLUTION ROUTINES - FPOWER SUBROUTINE

SUBROUTINE FPOWER (IV, LETHN)

C ROUTINE TO CALCULATE THE ACCURACY AND LETHALITY OF THE ARMAMENT
C SYSTEM SELECTED. THIS INCLUDES THE PROBABILITY OF HIT AS A FUNCTION
C OF RANGE, STATIONARY VS. MOVING FIRER & TARGET, AND ORDNANCE TYPE
C FOR THE SAME STANDARD TARGET. IF THERE ARE ANY CONSTRAINTS ON THESE
C VARIABLES SPECIFIED BY THE USER, THE PROGRAM TESTS FOR THEM AND IF
C THEY ARE NOT SATISFIED RETURNS AN INDICATOR OF WHICH COMPONENT TYPE
C TO BACK UP TO FOR SELECTING AN ALTERNATIVE CHOICE.

C INPUT PARAMETERS:
C IV VEHICLE
C
C OUTPUT PARAMETERS:
C LETHN RETURN CODE: 0 IF ALL CONSTRAINTS SATISFIED; OTHERWISE
C SET TO THE INDEX OF THE COMPONENT TYPE TO BACK UP TO.

COMMON /VHR/

DIMENSION IVECLE (72, 1), IVECLE (72, 1), ITURET (42, 9), PMAIN (44, 9),
+ FMAGN (40, 10), FMISGN (40, 7), IADGUN (40, 6), IRANGR (28, 7),
+ ISENSI (32, 9), ISTABLE (30, 6), IGNCTL (28, 9), IAMMO (30, 30),
+ IAMMOC (26, 9), IENGINE (40, 8), ITRANS (32, 8), IFINDR (28, 10),
+ ITROAD (30, 10), ISPRNG (30, 10), ITRACK (34, 8), ISKIRT (28, 9),
+ IFUEL (28, 10), IFUELC (26, 9), ICREW (26, 5), ICARGO (26, 5),
+ IEXT (26, 6), ICOMMO (28, 9), IFIREX (28, 10), IEVEN (28, 9),
+ IDTGN (26, 5), ISLGSN (26, 24), ISMOKE (26, 10), IEWSE (26, 10),
+ TRENH (3, 50), IVFILE (1), IVFILE (1)

EQUIVALENCE (VECLE (1), IVECLE (1)), (HULL (1), IHULL (1)), (TURRET (1),
+ ITURET (1)), (HAIN (1), PMAIN (1)), (MACGUN (1), FMAGN (1)),
+ (FMISGN (1), FMAGN (1)), (MISGN (1), FMISGN (1)), (ADGUN (1), IADGUN (1)), (RANGR (1),
+ IRANGR (1)), (SENSI (1), ISENSI (1)), (STABLE (1), ISTABLE (1)),
+ (IGNCTL (1), IGNCTL (1)), (AMMO (1), IAMMO (1)), (AMMOC (1),
+ IAMMOC (1)), (ENGINE (1), IENGINE (1)), (TRANS (1), ITRANS (1)),
+ (FIND (1), IFINDR (1)), (ROADH (1), ITROAD (1)), (SPRING (1),
+ SPRING (1)), (TRACK (1), ITRACK (1)), (SKIRT (1), ISKIRT (1)),


SOLUTION ROUTINES - FPOWER SUBROUTINE

C-81

+ (FUEL(1), IFUEL(1), (FUEL(1), IFUEL(1), (CREW(1),
+ ICREW(1), (CARGO(1), ICARGO(1), (ELECTR(1), IELECT(1),
+ (COMO(1), ICOMO(1), (FIREX(1), IFIREX(1), (ENVIRC(1),
+ TENVIR(1), (DIAGH(1), IDIAGH(1), SIGSUP(1), ISIGSP(1),
+ (SMOKES(1), ISMOKE(1), (EWSYS(1), TWESYS(1), (REIN(1), IRELN(1)

5 EQUIVALENCE (VEHICLE(1), VFILE(1),TVFILE(1))
6 COMMON /JPARS/
7 COMMON /JPARS1/
8 COMMON /NPARS/
9 COMMON /AUX/
10 REAL*8 BL9, DASHES
11 EQUIVALENCE (FNULL, NULL), (IBLS, BL5)
12 COMMON /ARRAYS/
13 REAL*8 MANUF, MODEL, NATION
14 DIMENSION ARRAY(1)
15 EQUIVALENCE (ARRAY(1), DETRNG(1))
16 COMMON /JPARS2/
17 COMMON /DATA/
18 COMMON /DATA2/

19 DIMENSION SS(2,3,3)
20 LOGICAL NORIESL
21 DATA KSTGT/1/, KMTGT/2/

C
22 IRETN = 0

C COMPUTE DISPERSION IN THE STATIONARY FIRER - STATIONARY TARGET CASE
C BY OBTAINING THE ROOT SUM OF SQUARES OF THE DISPERSIONS ASSOCIATED
C WITH EACH OF THE INDIVIDUAL COMPONENTS INVOLVED.
C FIRST SPT POINTERS TO COMPONENTS INVOLVED.
23 LMAING = IVECLE(JMAING, IV)
24 LMTSL = IVECLE(JMTSL, IV)
25 LRNGS = IVECLE(JRNGS, IV)
26 LSTBLS = IVECLE(JSTBLS, IV)
27 LCTLS = IVECLE(JCTLS, IV)
SOLUTION ROUTINES - FPPOWER SUBROUTINE

C LAMMO = IVECIE(JAMMO,IV)
C
C GET POINTERS TO DISPERSION ARRAYS ASSOCIATED WITH EACH COMPONENT
IARRMG = MAINGN(JSIGMA,LMAING)
IF (L4SL .NE. NULL) IARRML = MISGN(JSIGMA,LMTSL) + 8
IARRBG = IRANGR(JSIGMA,LNGS)Y
IARRCL = IGNCTL(JSIGMA,LCTL)S) + 3
C
C DETERMINE PERFORMANCE CLASS OF STABILIZATION SYSTEM
ISTBF = 1
IF (LSTBSY .NE. NULL) ISTBF = ISTATL(JPERF,LSTBSY) + 1
IF (ISTBF .LT. 1 .OR. ISTBF .GT. 3) CALL ERR(45,'FPPOWER',
+ ISTBF,1,3)
C
C DETERMINE WHETHER VEHICLE CAN FIRE MISSILES
NOMTSL = .TRUE.
C IS THERE A MISSILE LAUNCHER ON THE VEHICLE ?
IF (L4SL .EQ. NULL) GO TO 3
NOMTSL = .FALSE.
GO TO 10
C IS THERE A GUIDED MISSILE ORDNANCE THAT CAN BE FIRED FROM THE
C MAIN GUN ?
3 CAL = FMAING(JCALIB,LMAING)
IR = LAMMO
IF (IR .EQ. NULL) GO TO 10
IF (ABS(LAMMO(JCALIB,IR)-CAL) .GT. .001) GO TO 8
IF (LAMMO(JGUIDE,IR) .LE. 1) GO TO 8
NOMTSL = .FALSE.
GO TO 10
8 IR = LAMMO(JNEXT,IB)
GO TO 5
C
C ITERATE OVER RELEVANT ORDNANCE TYPES AND RANGES, COMPUTING ROOT
C SUM OF SQUARES OF COMPONENTS' DISPERSIONS.
1) DO 20 IORDTP=1,NORDTP
SOLUTION ROUTINES - FPPOWER SUBROUTINE

IF (IORDTP .EQ. NORDTP .AND. NOMSL) GO TO 20
IARR = IARRML
IF (IORDTP .EQ. NORDTP .AND. L.NOMSL .EQ. NULL) IARR = IARRML
DO 15 IRANGE=1, N RANGE
DO 15 IXY=1, IXY
SS(IXY, IRANGE, IORDTP) = SIGMA(IXY, IRANGE, IORDTP, IARR)**2
  + SIGMA2(IXY, IRANGE, IORDTP, IARRML)**2
  + SIGMA2(IXY, IRANGE, IORDTP, IARRCL)**2
15 CONTINUE

C CALCULATE THE PROBABILITY OF HIT AGAINST A STANDARD TARGET FOR
C VARIOUS RANGE, ORDNANCE, AND MOVING / STATIONARY CONDITIONS.

CALL INITVL (PHITS (1, 1, 1, IV), 18, FNULL)
CALL INITVL (PHITM (1, 1, 1, IV), 18, FNULL)
CALL INITVL (PK (1, 1, 1, IV), 18, FNULL)
DO 40 IORDTP=1, NORDTP
  IF (IORDTP .EQ. NORDTP .AND. NOMSL) GO TO 40
DO 35 IRANGE=1, N RANGE
  XSS2 = SS(1, IRANGE, IORDTP)
  YSS2 = SS(2, IRANGE, IORDTP)
  XSS = SQRT(XSS2)
  YSS = SQRT(YSS2)
  XMF2 = SIGMVF (1, IRANGE, IORDTP, ISTBPP)
  YMF2 = SIGMVF (2, IRANGE, IORDTP, ISTBPP)
  XMS = SQRT(XSS2 + XMF2)
  YMS = SQRT(YSS2 + YMF2)
  XMT2 = SIGMV (1, IRANGE, IORDTP, 1)
  YMT2 = SIGMV (2, IRANGE, IORDTP, 1)
  XSM = SQRT(XSS2 + XMT2)
  YSM = SQRT(YSS2 + YMT2)
  XMM = SQRT(XSS2 + XMF2 + XMT2)
  YMM = SQRT(YSS2 + YMF2 + YMT2)
PHITS (IRANGE, KSTGT, IORDTP, IV) = PHIT (XSS, YSS)
PHITS (IRANGE, KATGT, IORDTP, IV) = PHIT (XSM, YSM)
SOLUTION ROUTINES - POWER SUBROUTINE

ISMN
83 PHITM(IRANGE,KSTGT,IORDTP,IV) = PHIT(XMS,YMS)
81 PHITM(IRENGE,KSTGT,IORDTP,IV) = PHIT(XMM,YMM)

C CALCULATE THE PROBABILITY OF A KILL GIVEN A HIT AGAINST THE TARGET

82 DO 30 IASPEC = 1,NASPEC
83   PK(IRANGE,IASPEC,IORDTP,IV) = FVAL1(PKDAT(1,IRANGE,
      + IASPEC,IORDTP),RDISPR,NDISPR,(XSS+YSS)/2.)
84 30 CONTINUE
85 35 CONTINUE
86 40 CONTINUE
87 IVECLE(JPHITS,IV) = IV
88 IVECLE(JPHITM,IV) = IV
89 IVECLE(JPK,IV) = IV
90 RETURN
91 END
SOLUTION ROUTINES - GENVEH SUBROUTINE

TSN
1
SUBROUTINE GENVEH
C

ROUTINE TO GENERATE THE "BEST" CONCEPT VEHICLE FROM THE COMPONENTS
AVAILABLE AND MATCHING THE CONSTRAINTS SPECIFIED.
C

COMMON /VEH/
3
DIMENSION VECCE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
* + FMACN(40,10),PHSGH(40,7),IADGUN(40,6),IRANGR(28,7),
+ ISENSR(32,9),IESTABL(30,6),IGNCTRL(28,9),IAMMO(30,39),
+ YAMMOC(26,9),ISENGN(40,8),ITRANS(32,8),IFINDR(26,10),
+ IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKTR(28,8),
+ IFUEL(28,13),IFUEL(26,9),ICREW(26,5),ICARGO(26,5),
+ IELECT(26,6),ICOMMO(28,9),TIREX(28,10),ITENVIR(26,9),
* + IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSSYS(26,10),
+ IRELN(3,50),VFILE(1),IVFILE(1)

4
EQUIVALENCE (VEHICLE(1),VECECE(1)),(IHULL(1),IHULL(1)),(ITURET(1),
+ ITURET(1)),(MAINGN(1),FMING(1)),(MACGUN(1),FMACN(1)),
+ (PHSGH(1),IADGUN(1)),IADGUN(1)),(RANGR(1),
+ (IRANGR(1)),(SENSR(1),IESTABL(1)),(STABLE(1),IESTABL(1)),
+ (IANGCTRL(1)),IANGCTRL(1)),(IAMMO(1)),IAMMO(1)),(IAMMOC(1),
+ (IAMMOC(1)),(ISENGN(1),ITRANS(1)),(ITRANS(1)),
+ IROADW(1)),(ISPRNG(1)),(ITRACK(1)),(SKIRT(1),ISKIRT(1)),
* + (IFUEL(1)),IFUEL(1),IPUEL(1)),IFUEL(1),IPUEL(1),ICREW(1),
+ ICARGO(1),ICARGO(1),ELECTR(1),IELECT(1),
+ (COMMO(1),ICOMMO(1)),(TIREX(1),TIREX(1)),(ITENVIR(1),
+ IENVIR(1)),(IDIAGN(1)),(IDIAGN(1)),(ISIGSP(1)),ISIGSP(1),
+ ISMOKE(1),ISMOKE(1),IEWSSYS(1),IEWSSYS(1)),(IRELN(1),IRELN(1))

5
EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))

6
COMMON /VPROTO/
7
REAL*8 FLDNAM,HRCNAM,DFPAULT
8
COMMON /JPARS/
9
COMMON /JPARS1/
10
COMMON /NPARS/
11
COMMON /AUX/
SOLUTION ROUTINES - GENVEH SUBROUTINE

ISN
12     REAL*8 ELO, DASHES
13     EQUIVALENCE (FNULL, NULL), (IBLS, BLS)

C
14     DIMENSION JJ(35), JCT(35), NUM(35), MATCH(35), NUMSPC(35)
15     LOGICAL OKCMP, COMPAT

C
16     ORDER THE COMPONENTS OF EACH TYPE IN DECREASING ORDER BY SCORE
C
17     CALL CSCORE
C
18     DETERMINE NUMBER OF USER SPECIFICATIONS FOR EACH COMPONENT
C
17     CALL IZERO(NUMSPC, NCMPTP)
18     DO 2 ICMP=1, NCMPTP
19     IF (IVECLE(ICMP, ISPECV) .EQ. NULL) GO TO 2
20     IOFF = IPILPT(ICMP+1)
21     NUMSPC(ICMP) = NLSTL(VFILE(IOFF), NFLDS(ICMP+1), JNEXT, 
22     + IVECLE(ICMP, ISPECV))
22     2 CONTINUE

C
22     -- VEHICLE SPECIFICATION FILE CONTAINS INFORMATION
C
22     -- ABOUT THE VEHICLE MODEL WHICH INTERFERES WITH
C
22     -- THE MATCHING OF SPECIFICATION TO ALTERNATIVES COMPONENTS.
C
22     -- THEREFORE, THE JMODEL FIELD MUST BE NULL BEFORE
C
22     -- THE BEST VEHICLE CAN BE GENERATED.
C
23     DO 4 ICMP=1, NCMPTP
24     IR = IVECLE(ICMP, ISPECV)
25     3 IF (IR .EQ. NULL) GO TO 4
26     IOFF = IPILPT(ICMP+1)
27     NF = NFLDS(ICMP+1)
28     CALL STORE(NULL, IVFILE(IOFF), NF, JMODEL, IR)
29     IR = IVAL(IVFILE(IOFF), NF, JNEXT, IR)
30     GO TO 3
31     4 CONTINUE
C
C
SOLUTION ROUTINES - GENER SUBROUTINE

C ITRATE OVER COMPONENT TYPES IN ORDER THAT WISH TO PROCESS THEM
C
32   I = 1
33  5   ICMP = ICMPR (I)
34   IRECTP = ICMP + 1
35   NF = NFDLS (IRECTP)
36   IOFF = IFILPT (IRECTP)

C ITRATE OVER COMPONENTS OF THIS TYPE (THEY ARE IN APPROPRIATE ORDER
C ALREADY)
C
37   JJ = ICMPPT (ICMP)
38   JJ (I) = JJ
39   JCT (I) = 0
40   NUM (I) = 0
41   MATCH (I) = 0

C DOES COMPONENT JJ SATISFY USER SPEC'S?
C
42   IF (NUMSPC (ICMP) .EQ. 0 .AND. MINNUM (ICMP) .EQ. 0) GO TO 10
43   8   IF (NUMSPC (ICMP) .GT. 0 .AND.
44      +    .NOT. OKCMP (ICMP, JJ, VFILE (IOFF), NF, MATCH (I), NUM (I), NUMSPC (ICMP)))
45      +    GO TO 12

C IS COMPONENT JJ COMPATIBLE WITH PREVIOUSLY SELECTED COMPONENTS?
C
46   IF (.NOT. COMPAT (ICMP, JJ)) GO TO 15
C
C REMEMBER COMPONENT JJ AS PART OF SPECIFICATION VEHICLE
C
47   CALL QUEUE (JJ, IVECLE (ICMP, ICANDV), IRECTP)
C
C HAS MINIMUM NUMBER OF COMPONENTS OF THIS TYPE BEEN SELECTED?
C IF NOT, GO GET ANOTHER.
C
48   IF (NUM (I) .LT. MINNUM (ICMP)) GO TO 15
SOLUTION ROUTINES - GENVEH SUBROUTINE

C
C CALCULATE ANY PERFORMANCE PARS APPROPRIATE AFTER SELECTION OF THIS
C COMPONENT TYPE

47     IRETN = IPFORM(ICMP)
48     IF (IRETN .EQ. 0) GO TO 10
C
C IF BACKTRACKING IS INDICATED, DO SO

49     IF (IRETN .GE. 1) CALL ERR(44,'GENVEH',I,IRETN,ICMP)
50     DO 11 K=IRETN,1
51     CALL UNHOOK(ICMPK(K))
52     11 CONTINUE
53     I = IRETN
54     GO TO 22
C
C PROCESS NEXT COMPONENT TYPE IF THERE IS ONE

55    10     I = I + 1
56     IF (I .GT. NCMPTP) RETURN
57     GO TO 5
C
C COMPONENT DIDN'T SATISFY USER SPECS. KEEP NOTE OF NUMBER OF SUCH
C COMPONENTS.

58    12     JCT(I) = JCT(I) + 1
C
C GET NEXT COMPONENT OF CURRENT TYPE, IF THERE IS ONE

59    15     JI = IVAL(VFILE(IOFF),NP,JSCORE,J1)
60     JJ(I) = JI
61     IF (JI .NE. NULL) GO TO 8
C
C CAN FAILURE TO FIND A SUITABLE COMPONENT POSSIBLY BE AVOIDED BY
C BACKTRACKING ?

62     IF (JCT(I)+INNUNM(ICMP) .LE. NALTCS(ICMP)) GO TO 20
SOLUTION ROUTINES - GENVEN SUBROUTINE

63 CALL ERR(42,'GENVEN',ICMP,J1,JCT(I))
64 PAUSE

C BACKTRACK TO PREVIOUS COMPONENT TYPE
C
65 25 CALL UNHOOK(ICMP)
66 I = I - 1
67 22 IF (I .GE. 1) GO TO 25
68 CALL ERR(43,'GENVEN',ICMP,0,0)
69 PAUSE

C TRY NEXT CHOICE OF PREVIOUS COMPONENT TYPE
C
70 25 ICMP = ICMPK(I)
71 IRECTP = ICMP + 1
72 NF = NELDS(IRECTP)
73 IOFF = IFLIPT(IRECTP)
74 JI = TVAL(VFILE(IOFF),NF,JSCORE,JJ(I))
75 IF (JI .EQ. NULL) GO TO 20
76 NUM(I) = 1
77 CALL UNHOOK(ICMP)
78 MATCH(I) = 1
79 JJ(I) = JI
80 GO TO 8

C END

81 FUNCTION TPFORM(ICMP)

C ROUTINE TO SUPERVISE THE CALCULATION OF PERFORMANCE VARIABLES.
C THE ROUTINE IS CALLED AFTER EACH COMPONENT OF A GIVEN TYPE IS
C SELECTED FOR THE CANDIDATE VEHICLE. IF THIS IS AN APPROPRIATE TIME
C TO CALCULATE A GIVEN PERFORMANCE PARAMETER (I.E., ALL RELEVANT
C INDEPENDENT VARIABLES ARE KNOWN AT THIS POINT OR CAN BE ESTIMATED
C SUFFICIENTLY WELL TO TEST A CONSTRAINT ON A PERFORMANCE PARAMETER),
C THE ROUTINE DOES SO. WHENEVER THE ROUTINE CALCULATES A PERFORMANCE
C PARAMETER IT CHECKS WHETHER IT SATISFIES ANY CONSTRAINTS THAT MAY BE
SOLUTION ROUTINES - GENVEH SUBROUTINE

ISN

C IMPOSED ON IT, EITHER BY THE USER OR FROM THE PROGRAM'S KNOWLEDGE OF
C VEHICLE DESIGN. IF A PERFORMANCE PARAMETER DOES NOT MEET AN IMPOSED
C CONSTRAINT, THE CODE INDICATES HOW FAR TO BACKUP TO TRY TO OBTAIN A
C BETTER MATCH (I.E., WHAT COMPONENT TYPE TO RETURN TO TRY TO SELECT
C A DIFFERENT CHOICE). THE FUNCTION RETURNS THE INDEX OF SUCH A
C COMPONENT IF ANY TESTED CONSTRAINTS FAIL, OR IT RETURNS ZERO
C OTHERWISE.
C
C INPUT PARAMETERS:
C ICMP TYPE OF COMPONENT JUST SELECTED PRO CANDIDATE VEHICLE

83 COMMON /VEH/
84 DIMENSION IVECLE(72,3),IHULL(48,9),ITURRET(42,9),FMAING(44,9),
* FMACGN(40,10),FMISGN(40,7),IADGN(40,6),IAGR(28,7),
* ISENSR(32,9),ISTABL(33,6),ICT(28,9),IAMMO(30,3),
* IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
* IROADW(36,10),ISPRNG(34,10),ITRACK(34,8),ISKIRT(28,8),
* IFUEL(28,10),IFUELC(26,9),TCREW(26,5),TCARGO(26,5),
* ISELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
* IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IWEWSYS(26,10),
* IRELN(3,50),VFIL(1),IVFIL(1)
85 EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
* ITURRET(1)),(MATING(1),FMAING(1)),(MACGN(1),FMACGN(1)),
* (MISGN(1),FMISGN(1)),(ADGN(1),IADGN(1)),(RANGER(1),
* ITRANS(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
* (ICT(1),IGCT(1)),(TAMMO(1),IAMMO(1)),(AMMOC(1),
* IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANS(1),ITRANS(1)),
* (FINDR(1),IFINDR(1)),(ROADW(1),IROADW(1)),(SPRNG(1),
* ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
* (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
* ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),ISELECT(1)),
* (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIR(1),
* IENVIR(1)),(DIAGN(1),IDIAGN(1)),(SIGSP(1),ISIGSP(1)),
* (SMOKE(1),ISMOKE(1)),(WSYS(1),IWEWSYS(1)),(RELN(1),IRELN(1))
86 EQUIVALENCE (VEHICLE(1),VFIL(1),IVFIL(1))
87 COMMON /JPARS/

C-90
SOLUTION ROUTINES - GENVEH SUBROUTINE

COMMON /JPARS1/
COMMON /AUX/
REAL*8 DL8, DASHES
EQUIVALENCE (FNULL, NULL), (IBLS, BL5)
COMMON /SPECS/
REAL*8 ATRENG, ATRPER, VEHCAT, VEHCMP, VEHENG, VEHPER

C LOGICAL OKVAL
C
IPFCRM = 0
GO TO (10, 20, 10, 10, 10, 10, 10, 10, 10, 10, 80, 10, 100,
+ 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 160), ICMP
C
10 RETURN
C
HAVE JUST SELECTED TURRET. CAN NOW COMPUTE THE GEOMETRY, WEIGHT,
C PROTECTION, AND MANY OF THE MOBILITY PARAMETERS IN FINAL FORM.
C
20 CALL DIMENS (ICANDV, IPFORM)
C
RETURN
C
HAVE JUST SELECTED ARMAMENT COMPONENTS. COMPUTE ACCURACY &
C LEATHALITY PARAMETERS AND CHECK THAT THEY SATISFY ANY USER-PROVIDED
C CONSTRAINTS.
C
60 CALL FPPOWER (ICANDV, IPFORM)
C
RETURN
C
HAVE JUST SELECTED THE ENGINE. CAN MAKE QUICK TEST THAT HAVE
C SUFFICIENT GROSS HP / TON.
C
80 IF (VEHICLE(JGHTN, ISPECV) .EQ. FNULL) RETURN
C
103 LENGIN = IVECLE(JENGIN, ISPECV)
104 HP = ENGINE(JHP, LENGIN)
105 IF (OKVAL(HP/VEHWT, 1, JGHTN, NULL)) RETURN
106 IPFORM = ICMP
C
RETURN
C
HAVE JUST SELECTED THE FINAL DRIVE. CAN MAKE INITIAL ESTIMATE OF
SOLUTION ROUTINES - GENVEH SUBROUTINE

C POWER LIMITED MOBILITY PARAMETERS & BACKTRACK IF THESE DON'T MEET C CONSTRAINTS.
108 1)) CALL MOBILE (ICANDV,IPFORM)
109    RETURN
C HAVE SELECTED ALL COMPONENTS, CHECK RAM/D & COST CONSTRAINTS.
C (THIS PORTION NOT IMPLEMENTED).
110 160 RETURN
111    END
SOLUTION ROUTINES - MOBILE SUBROUTINE

SUBROUTINE MOBILE(IV, IRETN)

ROUTINE TO CALCULATE SELECTED POWER-CONSTRAINED MOBILITY
PARAMETERS TO SEE WHETHER THE CHOICE OF POWER TRAIN COMPONENTS IS
SATISFACTORY IF THE USER HAS SPECIFIED ANY CONSTRAINTS ON THESE
PARAMETERS.

INPUT PARAMETERS:
IV VEHICLE OF INTEREST

OUTPUT PARAMETERS:
IRETN COMPONENT RANK INDEX TO BACK UP TO IF THE USER'S
CONSTRAINTS ARE NOT SATISFIED; 0 OTHERWISE.

COMMON /VER/
DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
   FMCAN(40,10), FPHISN(40,7), IADGUN(40,6), ITANGR(28,7),
   IISENSR(32,9), IISTABL(30,6), IGNCTL(28,9), IAMMO(30,39),
   IIAMMOC(26,9), IENGIN(40,8), IITANS(32,8), IFINDR(28,10),
   ITROADW(36,10), ITSPEN(39,10), IITRACK(34,8), IISKIFT(29,8),
   IFUEL(28,10), IFUEL(26,9), ICHEW(26,5), ICARGO(26,5),
   IILECT(26,6), ICMMO(28,9), IFIREX(28,10), IIVVIR(26,9),
   ITDIAGN(26,5), ITSIGSP(26,24), ISMORS(26,10), IEWSYS(26,10),
   IRETN(3,50), VFILE(1), IVFILE(1)

EQUIVALENCE (VECLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
   ITURET(1)), (IENGIN(1), FMAING(1)), (MACGUN(1), FMCAN(1)),
   (MPHISN(1), FPHISN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
   ITANGR(1)), (ISENSR(1), IISENSR(1)), (STABLE(1), IISTABL(1)),
   (GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
   IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANSM(1), IITANS(1)),
   (FINLD(1), IFINDR(1)), (ROADW(1), ITROADW(1)), (SPRING(1),
   ISPRING(1)), (ITRACK(1), ITTRACK(1)), (ISKIFT(1), IISKIFT(1)),
   (FUEL(1), IFUEL(1)), (FUEL(1), IFUEL(1)), (CREW(1),
   ICHEW(1), ICARGO(1)), (ELECT(1), IILECT(1)),
   (COMMO(1), ICMMO(1)), (FIREX(1), IFIREX(1)), (ENVIR(1),
SOLUTION ROUTINES - MOBILE SUBROUTINE

C

LENVR (1), (DIAGN (1)), IDIAGN (1), (SIGSUP (1)), TSI GSP (1)),
+ (SMAKE (1)), ISMOKE (1)), (EWSYS (1)), IEWSYS (1)), (RLE (1), IREL (1))
EQUIVALENCE (VEHICLE (1), VFILE (1), IVFILE (1),
C
COMMON /VPRO/O/
RREAL*8 PLDNAM, RECNUM, DFAULT
COMMON /JPARS/
COMMON /JPARS 1/
COMMON /HPRS/
COMMON /AUX/
REAL*3 ELA, DASHES
EQUIVALENCE (FNULL, NULL), (IBLS, BL)
COMMON /SPECS/
REAL*8 ATRENG, ATRPER, VEHCAT, VEHNUM, VEHENG, VEHPER
COMMON /DATA/
COMMON /DATA 2/

C
LOGICAL OKVAL
C
IRETN = 0
C
LENIN = IVECLE(JENGIN, IV)
LTRANS = TVECLE(JTRANS, IV)
LFINDR = IVECLE(JFINDR, IV)

C CHECK GROSS HP / TON
HP = DVAL2 (ENGINE (JHP, LENIN), JENGIN, JHP, 'HP ')
HPTON = HP / VEHWT
IF (.NOT. OKVAL (HPTON, 1, JGHPNTN, NULL)) GO TO 999

C CHECK SPROCKET HP / TON
EFFTR = DVAL2 (TRANSM (JEFFIC, LTRANS), JTRANS, JEFFIC, ' - ')
EFFFD = DVAL2 (FILDR (JEFFIC, LFINDR), JFINDR, JEFFIC, ' - ')
SHPTON = HP * EFFTR * EFFFD / VEHWT
IF (.NOT. OKVAL (SHPTON, 1, JSHPNTN, NULL)) GO TO 999

C CHECK SPEED ON LEVEL ROAD IF CONSTRAINED
SOLUTION ROUTINES - MOBILE SUBROUTINE

ISN 30 IF (IVECLE(JMXSPD,ISPECV) = .EQ. NULL) GO TO 10
     SPDI = FVAL1(SPEED(1,1),RSHTN,NSHTN,SHPTON)
     IF (.NOT.OKVAL(SPDI,1,JMXSPD,NULL)) GO TO 999

C CHECK SPEED ON 30 DEGREE GRADE, IF CONSTRAINED
C 10 IF (IVECLE(J30SPD,ISPECV) = .EQ. NULL) GO TO 20
     SPD2 = FVAL1(SPEED(1,2),RSHTN,NSHTN,SHPTON)
     IF (.NOT.OKVAL(SPD2,1,J30SPD,NULL)) GO TO 999

C CHECK MAX SLOPE THAT TANK CAN CLIMB, IF THIS IS CONSTRAINED
20 IF (IVECLE(JSLOPE,ISPECV) = .EQ. NULL) GO TO 30
     SLOPMX = FVAL2(SLOPE,RSHTN,NSHTN,SHPTON,VEHWT,NVEHWT,VEHWT)
     IF (.NOT.OKVAL(SLOPMX,1,JSLOPE,NULL)) GO TO 999

C CHECK MAX ACCELERATION THAT VEHICLE CAN ACHIEVE, IF CONSTRAINED
30 IF (IVECLE(JACCEL,ISPECV) = .EQ. NULL) RETURN
     ACC = FVAL2(JACCEL,RSHTN,NSHTN,SHPTON,VEHWT,NVEHWT,VEHWT)
     IF (.NOT.OKVAL(ACC,1,JACCEL,NULL)) GO TO 999
     RETURN

C HAVE FOUND AT LEAST ONE POWER-LIMITED MOBILITY PARAMETER CONSTRAINT
C THAT WAS NOT SATISFIED. BACKUP ONE. TRY NEW FINAL DRIVE.
999 DO 1000 IRETN=1,ICMPTP
     IF (ICMPK(IRETN) = .EQ. JFINDR) RETURN
1000 CONTINUE
     PAUSE
     END
SOLUTION ROUTINES - FUNCTION OKCMP

ISN
1

LOGICAL FUNCTION OKCMP (ICMP, JI, IFILE, NF, MATCH, NUM, NUMSPC)

C
C ROUTINE TO TEST WHETHER A SELECTED COMPONENT IS COMPATIBLE WITH
C THE USER SPECIFICATIONS. THE SELECTED COMPONENT IS DEEMED COMPATIBLE
C WITH THE USER SPECIFICATIONS IF EITHER OF THE FOLLOWING CONDITIONS
C ARE MET: (1) THE USER HAS INDICATED A SPECIFIC COMPONENT TO BE
C SELECTED AND THE COMPONENT IN QUESTION IS THAT COMPONENT, AND
C (2) THERE IS AT LEAST ONE COMPONENT OF THE CURRENT TYPE THAT MAY BE
C SELECTED WITHOUT EXCEEDING THE MAX NUMBER OF COMPONENT CONSTRAINTS
C AND IF THERE ARE ANY SETS OF ENGINEERING CONSTRAINTS SPECIFIED
C WHICH HAVEN'T YET BEEN PAIRED WITH A SPECIFIC COMPONENT SELECTION,
C ONE SUCH SET CAN BE PAIRED WITH THE CURRENT SELECTION SUCH THAT NO
C CONSTRAINTS ARE VIOLATED. IN THESE CASES THE FUNCTION RETURNS A
C TRUE VALUE; OTHERWISE IT RETURNS FALSE.

C INPUT PARAMETERS:
C ICMP COMPONENT TYPE
C JI TOTAL COMPONENT TO BE TESTED FOR COMPATIBILITY
C IFILE FILE IN WHICH COMPONENTS ARE LOCATED
C NF NUMBER OF FIELDS IN RECORDS OF THIS FILE
C MATCH NUMBER OF MATCHES OBTAINED BETWEEN COMPONENTS OF THIS
C TYPE SPECIFIED BY THE USER AND COMPONENTS SELECTED SO FAR
C NUM NUMBER OF COMPONENTS OF THIS TYPE ALREADY SELECTED
C NUMSPC NUMBER OF COMPONENTS OF THIS TYPE REFERRED TO BY THE
C USER SPECIFICATIONS

C OUTPUT PARAMETERS:
C MATCH UPDATED BY 1 IF THE CANDIDATE COMPONENT CAN BE PAIRED
C WITH AN AS YET UNMATCHED USER SPECIFICATION
C NUM UPDATED BY 1 IF THE CANDIDATE COMPONENT IS ACCEPTABLE.

COMMON /VEH/

DIMENSION IVECLE (72, 3), IHULL (48, 9), ITURET (42, 9), FMAING (44, 9),
+ FMACGN (40, 13), PHISGN (40, 7), IADGUN (40, 6), IMARK (28, 7),
+ ISENSE (32, 9), ISTALL (37, 6), IHGCTL (28, 9), IAMMO (30, 39).
SOLUTION ROUTINES - FUNCTION OKCMP

ISN
+ TAMMOC (26, 9), IENGIN (40, 9), ITRANS (32, 8), IFINDR (28, 10),
+ IROADW (36, 10), ISPRNG (30, 10), ITRACK (34, 8), ISKIRT (28, 8),
+ IFUEL (29, 10), IFUCEL (26, 9), ICREW (26, 5), ICARGO (26, 5),
+ ISELECT (26, 6), ICOMMO (28, 9), IFIREX (28, 10), IENVIR (26, 9),
+ TDIAGN (26, 5), ILSIP (26, 24), ISMOKE (26, 10), IIEWSYS (26, 10),
+ IRELN (3, 50), VFILE (1), IVFILE (1)

4 EQUIVALENCE (VEHICLE (1), IVECLE (1)), (HULL (1), IHULL (1)), (TURRET (1),
+ ITURRET (1), (MAINGN (1), PMAING (1)), (MACGUN (1), PMACGUN (1)),
+ (MISGUN (1), PMISGUN (1)), (ADGUN (1), IADGUN (1)), (RANGER (1),
+ ITRANR (1)), (SENSOR (1), ISENSOR (1)), (STABLE (1), ISTABL (1)),
+ (GUNCTL (1), IGNCTL (1)), (AMMO (1), TAMMO (1)), (AMMOC (1),
+ TAMMOC (1)), (ENGINE (1), IENGIN (1)), (TRANSM (1), ITRANS (1)),
+ (FINLDR (1), IFINDR (1)), (ROADWH (1), IROADW (1)), (SPRING (1),
+ ISPRNG (1)), (TRACK (1), ITRACK (1)), (SKIRT (1), ISKIRT (1)),
+ (FUEL (1), IFUEL (1)), (FUCEL (1), IFUCEL (1)), (CREW (1),
+ ICREW (1)), (CARGO (1), ICARGO (1)), (ELECTR (1), ISELECT (1)),
+ (COMMO (1), ICOMMO (1)), (FIREX (1), IFIREX (1)), (ENVIR (1),
+ IENVIR (1)), (DIAGN (1), IDIAGN (1)), (SIGSUP (1), ISIGSUP (1)),
+ (SMOKF (1), ISMOKE (1)), (IIEWSYS (1), IIEWSYS (1)), (RELN (1), IRELN (1))

5 EQUIVALENCE (VEHICLE (1), VFILE (1), IVFILE (1))

6 COMMON /VPROTO/
7 REAL*9 FLDMAM, RECNAM, DEFAULT
8 COMMON /JPARS/
9 COMMON /JPARS1/
10 COMMON /NPAES/
11 COMMON /AUX/
12 REAL*8 BL8, DASHES
13 EQUIVALENCE (FNULL, NULL), (IBLS, BL8)

C EQUIVALENCE (JMATCH, JMODEL)
C USE AN UNUSED FIELD TO POINT FROM A USER SPEC RECORD TO THE
C ACTUAL COMPONENT PAIRED WITH IT.
C
14 DIMENSION IFILE(NFILE)
15 LOGICAL OKVAL, IKVAL
16 DATA NCMPMX/50/
SOLUTION ROUTINES - FUNCTION OKCMP

C
18 OKCMP = .TRUE.
19 IRECTP = ICMP + 1
C
C ARE THERE ANY USER SPECIFICATIONS?
20 IF (IVECLE(ICMP,ISPECV) .EQ. NULL) NUM=NUM+1
21 IF (IVECLE(ICMP,ISPECV) .EQ. NULL) RETURN
C
C IS THE CURRENT COMPONENT SPECIFICALLY REQUESTED BY THE USER?
22 IR = INLIST(IFILE(JID,J1),IFILE,NP,J1D,JNEXT,IVECLE(ICMP,ISPECV))
23 IF (IR .EQ. NULL) GO TO 10
24 MATCH = MATCH + 1
25 NUM = NUM + 1
26 IFILE(JMATCH,IR) = J1
27 RETURN
C
C ARE THERE ALREADY TOO MANY COMPONENTS?
28 10 IF (NUM + (NUMSPC - MATCH) .LE. MAXNUM(ICMP)) GO TO 12
29 CKCMP = .FALSE.
30 RETURN
C
C ITERATE OVER ANY UNMATCHED SETS OF ENGINEERING PARAMETER CONSTRAINTS
31 12 IR = IVECLE(ICMP,ISPECV)
C
C SKIP OVER ANY SPECIFICATIONS THAT REFER TO CONSTANT COMPONENTS
32 14 IF (IFILE(J1D,IR) .LT. NCMPMX) GO TO 30
C
C SKIP OVER ANY SPECIFICATIONS THAT HAVE ALREADY BEEN MATCHED TO
C ANOTHER COMPONENT
33 IF (IFILE(JMATCH,IR) .NE. NULL) GO TO 31
C
C DOES THE TRIAL COMPONENT SATISFY THIS SET?
34 IOFFR = IFILP(I1+1)
35 DO 20 IFLD=2,NP
36 IF (IFLD .EQ. J1D) GO TO 20
37 IVF = IVAL(IFILE(IOFFR),NP,IFLD,IP)
38 IF (IVF .EQ. NULL) GO TO 20
SOLUTION ROUTINES - FUNCTION OKCMP

C HAVE FOUND A CONSTRAINT. TEST IT. FIRST DETERMINE TYPE OF FIELD.
KFLD = IFLDPT(IRECTP) + IFLD
ITYP = IFLDTP(KFLD)
C CONSTRAINTS ON ARRAYS OR CHARACTER FIELDS ARE NOT IMPLEMENTED
IF (ITYP .EQ. ITYPES(3)) .OR. ITYP .EQ. ITYPES(4)) GO TO 20
C IS TYPE REAL?
IF (ITYP .NE. ITYPES(1)) GO TO 15
C IF SO GET VALUE
V = DVAL2(IFILE(IFLD,JI),ICMP,IFLD,UNITS(KFLD))
IF (.NOT.OKVAL(V,IRECTP,IFLD,IR)) GO TO 30
GO TO 20
C TYPE IS INTEGER
IV = IFILE(IFLD,JI)
IF (IV .EQ. NULL) IV = DEFAULT(KFLD)
IF (.NOT.IOKVAL(IV,IRECTP,IFLD,IR)) GO TO 30
GO TO 20
C COMPONENT DOES INDEED SATISFY THIS SET OF ENGINEERING CONSTRAINTS
NUM = NUM + 1
MATCH = MATCH + 1
IFILE(JMATCH,IR) = JI
RETURN
C
C GET NEXT USER SPECIFICATION
IR = IFILE(JNEXT,IR)
IF (IR .NE. NULL) GO TO 14
C
C COMPONENT INCOMPATIBLE WITH ENGINEERING CONSTRAINTS
OKCMP = .FALSE.
RETURN
END
SOLUTION ROUTINES - PHIT FUNCTION

**FUNCTION PHIT(XSIG,YSIG)**

C
C ROUTINE TO CALCULATE THE PROBABILITY OF A HIT AGAINST A TARGET OF
C A STANDARD SIZE AND SHAPE HAVING A SPECIFIED AIM POINT, GIVEN A
C STANDARD DEVIATION OF THE FIRE IN THE X AND Y DIRECTIONS. THE
C DISPERSION OF FIRE IS ASSUMED TO FOLLOW A BIVARIATE NORMAL
C DISTRIBUTION. THIS VERSION OF THE ROUTINE ASSUMES A SQUARE TARGET
C 2.34 M ON A SIDE WITH THE AIM POINT IN THE MIDDLE. THE STANDARD
C DEVIATIONS OF THE FIRING DISTRIBUTION IS ALSO ASSUMED TO BE
C EXPRESSED IN METERS.

DATA RSQRT2/.70711/,XGT/1.17/,YGT/1.17/

IF (XSIG .LT. 1.E-6) XSIG = 1.E-6
IF (YSIG .LT. 1.E-6) YSIG = 1.E-6
PHIT = ERF(RSQRT2*XTGT/XSIG) * ERF(RSQRT2*YTGT/YSIG)
RETURN
END
C.5: OUTPUT ROUTINES

This section contains listings of routines used to format and write the output tables which display the results of a model run.
OUTPLT ROUTINES - OUTARR SUBROUTINE

SUBROUTINE OUTARR(Arr, IRECTP, IFLD)

    ROUTINE TO OUTPUT THE VALUES ASSOCIATED WITH AN ARRAY WHICH
    IS THE VALUE OF A RECORD'S FIELD

    INPUT PARAMETERS:
    Arr       ARRAY OF INTEREST
    IRECTP    RECORD TYPE OF RECORD POINTING TO ARRAY
    IRECF     RECORD POINTING TO ARRAY
    IFLD      FIELD OF RECORD POINTING TO ARRAY

    COMMON /VPROTO/
    REAL*8 FLDNAM, RECNAM, DEFAULT
    COMMON /NPARS/
    COMMON /AUX/
    REAL*8 BLAS, CACHES
    EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
    COMMON /ARRAYS/
    REAL*8 MANUF, MODEL, NATION
    DIMENSION ARRAY(1)
    EQUIVALENCE (ARRAY(1), DETPNG(1))
    COMMON /JPARRS/

    DIMENSION Arr(1), ILEVS(3), NLEVS(3)

    GET PARAMETERS OF INTEREST ASSOCIATED WITH ARRAY

    ND = 0
    ND 1) I=1, NADIMS
       IF (IARDEF(IRECTP, I) .NE. IRECTP .OR. IARDEF(JFLDRC, I) .NE.
       + IFLD) GO TO 10
       ND = ND + 1
       NLEVS(ND) = IARDEF(JNLEV, I)
       ILEVS(ND) = IARDEF(JLEVPT, I)
    10 1) CONTINUE
    11 1) IFLDRC = IFLDCPT(IRECTP) + IFLD

C-102
OUTPUT ROUTINES - OUTARR SUBROUTINE

ISN 22
UNIT = UNITS(IFLCRC)

C BRANCH ACCORDING TO NUMBER OF DIMENSIONS OF ARRAY
23 IF (ND .LE. 0 .OR. ND .GT. 3) CALL ERR(38, 'OUTARR', WD, 0, 0)
24 GO TO (20, 40, 60, 0)

C 1-DIMENSIONAL ARRAY. ITERATE OVER LEVELS OF ARRAY, OUTPUTTING
C EACH WITH ASSOCIATED VALUES
25 N = NLEV5(1)
26 DO 30 I=1,N
27 K = ILEVS(1) + I - 1
28 IF (ARR(I) .EQ. FNULL) GO TO 31
29 WRITE (7,100) (ARRLAB(J,K), J=1,5), ARR(I), UNIT
30 GO TO 30
31 100 FORMAT (4X,5A4,10X,F10.2,4X,A4)
32 31 WRITE(7,101) (ARRLAB(J,K), J=1,5)
33 101 FORMAT (4X,5A4,16X, 'N/A')
34 CONTINUE
35 RETURN

C 2-DIMENSIONAL ARRAY. ITERATE OVER DIMENSIONS IN REVERSE ORDER.
C ITERATE OVER LEVELS OF ARRAY, OUTPUTTING EACH WITH ASSOCIATED VALUE
36 L = 0
37 N2 = NLEV5(2)
38 DO 50 JD=1,N2
39 K2 = ILEVS(2) + JD - 1
40 WRITE (7,100) (ARRLAB(J,K2), J=1,5)
41 N1 = NLEV5(1)
42 DO 50 IO=1,N1
43 L = L + 1
44 K1 = ILEVS(1) + IO - 1
45 IF (ARR(I) .EQ. FNULL) GO TO 51
46 WRITE (7,105) (ARRLAB(J,K1), J=1,5), ARR(L), UNIT
47 GO TO 50
48 105 FORMAT (7X,5A4,7X,F10.2,4X,A4)
49 51 WRITE(7,106) (ARRLAB(J,K1), J=1,5)
OUTPLT ROUTINES - OUTARR SUBROUTINE

50  106  FORMAT(7X,5A4,13X,'N/A')
51     50  CONTINUE
52  RETURN
C
C  3-DIMENSIONAL ARRAY. ITERATE OVER DIMENSIONS IN REVERSE ORDER.
C  ITERATE OVER LEVELS OF ARRAY, FOR EACH OUTPUTTING LABEL & VALUE
53  60  L = 0
54  54  N3 = NLEVS(3)
55     DO 70 KD=1,N3
56      K3 = ILEVS(3) + KD - 1
57      WRITE (7,100) (ARRLAB(J,K3),J=1,5)
58  58  N2 = NLEVS(2)
59     DO 70 JD=1,N2
60      K2 = ILEVS(2) + JD - 1
61      WRITE (7,105) (ARRLAB(J,K2),J=1,5)
62     DO 70 ID=1,N1
63      L = L + 1
64     IF(ARR(L) .EQ. FNULL) GO TO 71
65      K1 = ILEVS(1) + ID - 1
66      WRITE (7,110) (ARRLAB(J,K1),J=1,5),ARR(L),UNIT
67     GO TO 70
68  110  FORMAT (10X,5A4,4X,F10.2,4X,A4)
69  70    WRITE(7,111) (ARRLAB(J,K1),J=1,5)
71  111  FORMAT(10X,5A4,10X,'N/A')
72  70    CONTINUE
73  RETURN
74  END
OUTPLT ROUTINES - OUTCOM SUBROUTINE

ISN

1

SUBROUTINE OUTCOM

C ROUTINE TO OUTPUT THE ATTRIBUTES IN COMMON TO ALL COMPONENTS OF
C THE VEHICLE AS WELL AS COMBINED MEASURES OF THE VEHICLE AS A WHOLE.

2 COMMON /VPROTO/
3 REAL*8 FLONAM, RECNAME, DEFAULT
4 COMMON /JPARS/
5 COMMON /JPARS1/
6 COMMON /NPARS/
7 COMMON /OUTPLT/
8 LOGICAL*1 DF
9 REAL*8 XNAT, XMANUF, XMODEL

10 DIMENSION ISUB(3), CV(5)
11 DATA ISUB/'A', 'B', 'C'/, CV/ 'C', 'ONCE', 'PT V', 'EHIC', 'LE'/

C PRODUCE TABLE IA: VEHICLE COMPONENT IDENTIFICATIONS & PRODUCIBILITY
C OUTPUT TABLE TITLE
12 WRITE (7, 100) ISUB(1)
13 100 FORMAT ('1', '2X', 'TABLE IA', 'A1', ' CONCEPT VEHICLE: ',
14 + ' ATTRIBUTES IN COMMON TO ALL COMPONENTS.')

C OUTPUT TABLE COLUMN HEADERS
15 WRITE (7, 105)
16 105 FORMAT ('-', '2X', 'IDENTIFICATION', '--------- ', '10X,'/
17 + 'PRODUCTIBILITY', 'SUBSYSTEM / COMPONENT', '10X', 'NATION', ',',
18 + 'MANUF MODEL COMP INDEX # PROD RED TIME')

C ITERATE OVER COMPONENT LABELS, WRITING LABELS & ASSOCIATED VARIABLES
19 LAST = IPPARS - 1
20 KK = 0
21 DO 20 I = 1, LAST
22 IF (IOUTF(I) .NE. 0) GO TO 10
OUTPUT ROUTINES - OUTCOM SUBROUTINE

C WRITE SPACER OR SUBSYSTEM HEADER WITHOUT CORRESPONDING VALUES
20 WRITE (7,110) (OUTNAM(J,I),J=1,7)

21 110 FORMAT (1X,7A4)
22 GO TO 20

C WRITE LABEL & VALUES
23 10 KK = KK + 1
24 K = KOUT(KK)
25 WRITE (7,115) (OUTNAM(J,I),J=1,7),XNAT(K),XMANUF(K),XMODEL(K),
+ INDEX(K),NPROD(K),DF(1,K),RDTIME(K),DF(2,K)
26 115 FORMAT (1X,7A4,3X,A8,2(2X,A8),4X,I2,6X,I6,A1,2X,F6.1,A1)
27 20 CONTINUE

C OUTPUT SEPARATOR LINE & VEHICLE SUMMARY
C WRITE (7,120) CV,NPROD(NCMPT1)
28 120 FORMAT (82X,'_________/1X,5A4,60X,F6.1)

C NOW DO TABLE IB: NUMBER, WEIGHT, VOLUME, & RAM/D
C
C OUTPUT TABLE TITLE
29 WRITE (7,100) ISUB(2)

C OUTPUT TABLE COLUMN HEADERS
30 WRITE (7,125)
31 125 FORMAT ('**-47X,'WEIGHT',9X,'INTERNAL VOLUME',6X,'---------',
+ 'RAM/D -------/1X,' SUBSYSTEM / COMPONENT',10X,
+ 'NUMBER',7X,'LBS PERCENT',6X,'CU FT PERCENT',5X,
+ 'MAINT/OP MATURITY COMPLEXITY MMBF')

C ITERATE OVER COMPONENT LABELS, WRITING LABELS & ASSOCIATED VARIABLES
32 KK = 0
33 DO 40 I=1,LAST
34 IF (IOUTF(I) .NE. 0) GO TO 30
C WRITE SPACER OR SUBSYSTEM HEADER WITHOUT CORRESPONDING VALUES
35 WRITE (7,110) (OUTNAM(J,I),J=1,7)
36 GO TO 40
C WRITE LABEL & VALUES
ISN
37  30  KK = KK + 1
38  K = KOUT(KK)
39     IF( K .EQ. JHULL .OR. K .EQ. JTURET) GO TO 35
40     WRITE (7,130) (OUTNAM(J,I),J=1,7),NUMBER(K),WEIGHT(K),
       + DF(3,K),PCWT(K),VOLUME(K),DF(4,K),PCVOL(K),DWWUP(K),
       + DF(5,K),MATUR(K),DF(6,K),ICMPLX(K),DF(7,K),RELIAB(K),DF(8,K)
41          GO TO 40
42  35  WRITE(7,131) (OUTNAM(J,I),J=1,7),NUMBER(K),WEIGHT(K),
       + DF(3,K),PCWT(K),VOLUME(K),DF(4,K),DWWUP(K),
       + DF(5,K),MATUR(K),DF(6,K),ICMPLX(K),DF(7,K),RELIAB(K),DF(8,K)
43  130  FORMAT (1X,7A4,5X,12,4X,F9.2,A1,3X,F5.1,5X,F6.1,A1,3X,F5.1,5X,+
       + F6.1,6X,T1,A1,8X,I1,A1,3X,F7.1,A1)
44  131  FORMAT (1X,7A4,5X,12,4X,F9.2,A1,3X,F5.1,5X,F6.1,A1,3X,'7/AV/A',5X,+
       + F6.1,A1,6X,I1,A1,8X,I1,A1,3X,F7.1,A1)
45  40  CONTINUE
C  WRITE MISC SYSTEM LINE
C  WGT&M = FMCWGT * WEIGHT(NCMPT1)
46  VOLL&M = SUM(VOLUME(JMAING),NCMPT1-3)
47  VOLL&M = VOLUME(NCMPT1) - VOLL&M
48  IF (VOLL&M .LE. 0.0) VOLL&M = 0.0
49  PCT11 = VOLL&M / VOLUME(NCMPT1) * 100.
50  WRITE(7,132) WGT&M,FMCWGT,VOLL&M,PCT11
51  132  FORMAT(1X,'9. MISC SYSTEMS/ COMPS ','5X,'0*','4X,F9.2,1X,+
       + 3X,F5.1,5X,F6.1,1X,3X,F5.1,5X,'0.0*','1X,6X,'0*','9X,'0*',+
       + 3X,'0.0*')
C OUTPUT SEPARATOR LINE & VEHICLE SUMMARY
52  WRITE(7,135) CV,WEIGHT(NCMPT1),PCWT(NCMPT1),VOLUME(NCMPT1),+
       + PCVCL(NCMPT1),DWWUP(NCMPT1),MATUR(NCMPT1),ICMPLX(NCMPT1),+
       + RELIAB(NCMPT1)
53  135  FORMAT (34X,'__','9X,____,4X,____,6X,____,4X,____,7X,+
       + ______,6X,___,8X,__,6X,____/1,5A4,14X,'1*','4X,F9.2,4X,+
       + F5.1,5X,F6.1,4X,F5.1,5X,F6.1,7X,I19X,I1,4X,F7.1)
54  RETURN
55  END
OUTPUT ROUTINES - OUTENG SUBROUTINE

ROUTINE TO OUTPUT ENGINEERING PARAMETERS FOR A SPECIFIED VEHICLE

INPUT PARAMETERS:
IV - DESCRIPTION OF VEHICLE OF INTEREST

COMMON /VFH/
DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9), + FMACGN(4), I4, FMSGN(4), I7, IADGUN(4), I6, I4RANG(28,7), + ISENPR(32,9), I4STABL(30,6), I4GNCTL(28,9), I4AMMO(30,30), + I4AMMOC(26,9), I4ENGIN(40,8), I4TRANS(32,8), IIFINDR(28,10), + I4ROADW(36,10), I4SPRNG(30,17), I4TRACK(34,8), I4SKIRT(28,8), + I4FUEL(28,10), I4FUEL(26,9), I4CREW(26,15), I4CARGO(26,5), + I4ELECT(26,6), I4CMMO(28,9), I4FIREX(28,10), I4ENVIR(26,9), + I4IDIAGN(26,5), I4ISGSP(26,24), I4SNOKE(26,10), I4EWSYS(26,10), + I4RELN(3,50),VFIE(1), IVFILE(1)

EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1), + ITURET(1)), (MAING(1), FMAING(1)), (MACGN(1), FMACGN(1)), + (MSGN(1), FMSGN(4)), (ADGN(1), IADGUN(4)), (RANGER(1), + IRANG(1)), (SENSOR(1), I4SENPR(1)), (STABL(1), I4STABL(1)), + (GUNCTL(1), I4GNCTL(1)), (AMO(1), I4AMMO(1)), (AMMOC(1), + I4AMMOC(1)), (ENGINE(1), I4ENGIN(1)), (TRANS(1), I4TRANS(1), + (FINHR(1), I4IFINDR(1)), (ROADW(1), I4ROADW(1)), (SPRNG(1), + (ICREW(1), I4FUEL(1)), (FUEL(1), I4FUEL(1)), (FUEL(1), I4FUEL(1)), (CREW(1), + ICREW(1)), (CAPGO(1), I4CARGO(1)), (ELECT(1), I4ELECT(1)), + (COMM(1), I4COMM(1)), (FIREX(1), I4FIREX(1)), (ENVIR(1), + (ENVIR(1)), (DIAGN(1), I4IDIAGN(1)), (SIGS(1), I4SIGS(1)), + (SMOKE(1), I4SNOKE(1)), (EWSYS(1), I4EWSYS(1)), (RELN(1), I4RELN(1))

EQUIVALENCE (VEHICLE(1), VFIE(1), IVFILE(1))

COMMON /VPROTO/
REAL*8 FDONAM, RECNUM, DEFAULT
COMMON /JPARS/
COMMON /JPARS1/
OUTPUT ROUTINES - OUTENG SUBROUTINE

ISN
10 COMMON /NPARS/
11 COMMON /AUX/
12 REAL*8 BLB,DASHES
13 EQUIVALENCE (FNULL,NULL),(IBLS,PLS)
14 COMMON /ARRAYS/
15 REAL*8 MANUF,MODEL,NATION
16 DIMENSION ARRAY(1)
17 EQUIVALENCE (ARRAY(1),DETRNG(1))
18 COMMON /JPARS2/
19 LOGICAL*1 BL,DF
20 DATA BL/* * /

C OUTPUT TABLE TITLE
21 WRITE (7,100)
22 100 FORMAT ('11', ' TABLE 2. CONCEPT VEHICLE, ENGINEERING', ' + ' PARAMETERS/', ' ')

C WRITE FIRST COMPONENT NAME
C
23 WRITE(7,105) (OUTNAM(J,ICMATR),J=1,7)

C ITERATE OVER ENGINEERING PAR LABELS
24 LAST = LSTOUT
25 IFRST = IEPARS
26 I = IEPARS
27 IF (IOUTF(I) .NE. 0) GO TO 8

C TEST IF ANOTHER COMPONENT ON THE VEHICLE

28 IF(I .EQ. IFRST) GO TO 5
29 INST = IVAL(IVTLE(IOFF),NF,JNEXT,IREC)
30 IF(INST .EQ. NULL) GO TO 5
31 IVECLE(ICMP,IV) = INST
32 I = IFRST
33 GO TO 2
OUTPUT ROUTINES - OUTENG SUBROUTINE

C WRITE SPACER OR HEADER WITHOUT CORRESPONDING VALUE
34 5   WRITE (7,105) (OUTNAM(J,1),J=1,7)
35 105 FORMAT (1X,7A4)
36     IFIRST = 1
37     GO TO 80
C
C GET VALUE ASSOCIATED WITH LABEL
38  8   IFLDRC = IOUTF(1)
39   ITYPE = IFLDTP(IFLDRC)
40   IFLD = IFLDVL(IFLDRC)
41   IRECPT = IFLRCRC - IFLD
42   IRECTP = ITABLE(IFLDPT,IRECPT,NRECTP,1)
43   NF =NFLDS(IFRECTP)
44   IOFF = IFILPT(IFRECTP)
45   ICMP = IRECTP - 1
46   IREC = IVECLE(ICMP,IV)
47 IF (IREC.EQ. NULL) GO TO 80
48    DO 10 K=1,NTYPES
49  10   IF (ITYPES(K).EQ. ITYPE) GO TO 15
50  10   CONTINUE
51    CALL ERR(10,'OUTENG','VEHIC',FLDNUM(IFLDRC),ITYPE)
52  15   UNIT = UNITS(IFLDRC)
53   DF = BL
C BRANCH ACCORDING TO CATATYPE
54     GO TO (20,30,40,50),K
C
C REAL VALUE
55  20   V = DVAL(VFILE(IOFF),NF,IFLD,IREC,IRECTP,UNIT,DF)
56    WRITE (7,110) (OUTNAM(J,1),J=1,7),V,DF,UNIT
57  110  FORMAT (1X,7A4,1X,F10.2,A1,3X,A4)
58     GO TO 80
C
C INTEGER VALUE
59  30   IVL = IDVAL(VFILE(ICFF),NF,IFLD,IREC,IRECTP,UNIT,DF)
60    WRITE (7,115) (OUTNAM(J,1),J=1,7),IVL,DF,UNIT
61  115  FORMAT (1X,7A4,3X,I8,A1,3X,A4)
OUTPUT ROUTINES - OUTENG SUBROUTINE

I SN
62 GO TO 80
C
C REAL ARRAY
63 40 IOFF = IFILPT(IRECTP)
64 IVL = IVAL(VFILE(IOFF),NF,IFLD,IREC)
65 IF (IVL .EQ. NULL) GO TO 20
C GET POINTER TO ARRAY DESCRIPTION
66 DO 42 IARR=1,NARRS
67 IF (IARDF2(JRECTP,IARR) .EQ. IRECTP .AND. IARDF2(JFLDRC,
70 + IARR) .EQ. IFLD) GO TO 44
68 42 CONTINUE
C NUMBER OF ELEMENTS IN ARRAY
69 44 NV = IARDF2(JARSIZ,IARR)
C OFFSET IN FILE CONTAINING ARRAYS
70 IOFFAR = IARDF2(JARRPT,IARR)
C GET OFFSET OF ARRAY TO BASE
71 IOFF = IOFFAR + NV*(IVL-1)
C WRITE CONTENTS OF ARRAY
72 WRITE(7,105) (OUTNAM(J,I),J=1,7)
73 CALL CUTFARR(ARRAY(IOFF),IRECTP,IFLD)
74 GO TO 80
C
C CHARACTER VALUE (SHOULDN'T BE ANY)
75 50 CONTINUE
76 80 I = I + 1
77 IF (I .LE. LAST) GO TO 2
78 RETURN
79 END
OUTPLT ROUTINES - OUTPER SUBROUTINE

SUBROUTINE OUTPER(IV)

ROUTINE TO OUTPUT PERFORMANCE PARAMETERS FOR A SPECIFIED VEHICLE

INPUT PARAMETERS:

IV DESCRIPTION OF VEHICLE OF INTEREST

COMMON /VEH/
DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
+ FMAGN(40,9), FMSGN(40,7), IADGUN(40,6), IRANGR(28,7),
+ ITNSR(32,9), ISTABL(30,6), IGNLTL(28,9), IAMMO(30,30),
+ IAMMOC(26,9), IENGIN(40,8), ITNSR(30,5), IFINDR(28,10),
+ IRODEW(36,10), ISPRNG(33,13), ITRACK(34,8), ISKIRT(28,8),
+ IFUEL(28,10), IFUCLC(26,9), ITCREW(26,5), ICARGO(26,5),
+ IVELECL(26,6), ICMCMO(28,9), IFIREX(28,10), IENVIR(26,9),
+ ITDIAGN(26,5), TISGSP(26,24), ISMOKE(26,10), IWSYS(26,10),
+ ITRELN(3,50), FILE(1), IVFILE(1)

EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (SURET(1),
+ ITIPET(1)), (MAING(1), FMAING(1)), (MACUN(1), FMACU(1)),
+ (MSIGN(1), FNSGN(1)), (ADGUN(1), IADGUN(1)), (RANGR(1),
+ IRANGR(1)), (SENSR(1), ISENSR(1)), (STABLE(1), ISTABL(1)),
+ (GNTCL(1), IGNTLC(1)), (AMMOC(1), IAMMO(1)), (AMMOC(1),
+ IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANSM(1), ITRANS(1),
+ (FINDL(1), IFINDR(1)), (RODEWH(1), IRODEW(1)), (SPRNG(1),
+ ISPNGN(1)), (ITRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1),
+ (FUEL(1), IFUEL(1)), (FUEL(1), IFUEL(1)), (CREW(1),
+ ITCREW(1)), (ICARGO(1), ICARGO(1)), (ELECTR(1), IEELECTR(1)),
+ (CMCMO(1), CMCMO(1)), (IFIREX(1), IFIREX(1)), (EERRIR(1),
+ IENVIR(1)), (TDIAGN(1), TDIAGN(1)), (SIGSP(1), SIGSP(1)),
+ (TSMOKE(1), ISMOKE(1)), (IWSYS(1), IWSYS(1)), (TRELN(1), TRELN(1))

EQUIVALENCE (VEHICLE(1), FILE(1), IVFILE(1))

COMMON /VPROTO/
REAL*8 FLDNAM, RECNAM, DFAULT
COMMON /NPARS/
COMMON /AUX/
OUTPUT ROUTINES - OUTPER SUBROUTINE

ISN
10   REAL*8 BL8,CASHES
11   EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
12   COMMON /ARRAYS/
13   REAL*8 MANUF,MODEL,NATICN
14   DIMENSION ARRAY(I)
15   EQUIVALENCE (ARRAY(I),DETRNG(I))
16   COMMON /JPARS2/
C
17   LOGICAL*1 BL,DF
18   DATA BL/.T./
C
19   IRECTP = 1
C
C OUTPUT TABLE TITLE
20   WRITE (7,100)
21  100  FORMAT ('1',1X, 'TABLE 2. CONCEPT VEHICLE, PERFORMANCE', +
           'PARAMETERS.')
C
C ITERATE OVER PERFORMANCE PAR LABELS
22   LAST = ICMATR - 1
23   NF = NFLDS(1)
24   DO 80 I=IPPARS,LAST
25      IF (IOUTF(I),NE. 0) GO TO 8
26   5    WRITE (7,105) (OUTNAM(J,I),J=1,7)
27  105   FORMAT (1X,7A4)
28   GO TO 80
C
C GET VALUE ASSOCIATED WITH LABEL
29   IFLD = IOUTF(I)
30   ITYPE = IFLDTP(ILFD)
31   DO 10 K=1,NTYPES
32      IF (ITYPES(K),EQ.,ITYPE) GO TO 15
33  10     CONTINUE
34   CALL EPR(10,'OUTPER','VEHIC',FLDNAM(ILFD),ITYPE)
35  15     UNIT = UNITS(ILFD)
OUTPUT ROUTINES - OUTPER SUBROUTINE

36  DF = RL
   C  BRANCH ACCORDING TO CATATYPE
37    GO TO (20,33,4),50,K
   C
   C  REAL VALUE
38  20    V = DVAL(IVECLE,NF,IFLD,IV,UNIT,DF)
39    WRITE (7,110) (OUTNAM(J,I),J=1,7),V,DF,UNIT
40  110   FORMAT (1X,7A4,1X,F10.2,A1,3X,3A4)
41    GO TO 80
   C
   C  INTEGER VALUE
42  30    IVL = IDVAL(IVECLE,NF,IFLD,IV,UNIT,DF)
43    WRITE (7,115) (OUTNAM(J,I),J=1,7),IVL,DF,UNIT
44  115   FORMAT (1X,7A4,3X,18,A1,3X,A4)
45    GO TO 80
   C
   C  REAL ARRAY
46  40    IOFF = IFILPT(IRECTP)
47    IVL = IVAL(IVFILE(IOFF),NF,IFLD,IV)
48    IF (IVL .EQ. NULL) GO TO 20
   C  GET POINTER TO ARRAY DESCRIPTION
49    DO 42 IARR=1,NARRS
50       IF (IARDF2(IRECTP,IARR) .EQ. IRECTP .AND. IARDF2(JFLDRC,
      +               IARR) .EQ. IFLD) GO TO 44
51  42    CONTINUE
   C  NUMBER OF ELEMENTS IN ARRAY
52  44    NV = IARDF2(JARPSIZ,IARR)
   C  OFFSET IN FILE CONTAINING ARRAYS
53    IOFFAR = IARDF2(JARRPT,IARR)
   C  GET OFFSET OF ARRAY TO BASE
54    IOFF = IOFFAR + NV*(IVL-1)
   C  WRITE CONTENTS OF ARRAY
55    WRITE(7,105) (OUTNAM(J,I),J=1,7)
56    CALL CUTFARR(ARRAY(IOFF),IRECTP,IFLD)
57    GO TO 80
   C
OUTPUT ROUTINES - OUTVEH SUBROUTINE

SUBROUTINE OUTVEH

ROUTINE TO OUTPUT BEST CANDIDATE VEHICLE

COMMON /VEH/

DIMENSION IVECLE(72,3),IHULL(48,9),ITURRET(42,9),FMAING(44,9),
+ FMACGN(40,10),FMSGN(40,7),IADGUN(40,6),IRANGR(28,7),
+ ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
+ IAMMOC(26,9),IENGIN(4,8),ITRANS(32,8),IFINDR(28,13),
+ ITROWD(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
+ IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
+ IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IFENVIR(26,9),
+ IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
+ IRELN(3,50),VFIE(1),IVFIE(1)

EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
+ ITURRET(1)),(MAINGN(1),FMAING(1)),(MACGN(1),FMACGN(1)),
+ (MISGN(1),FMSGN(1)),(IADGUN(1),IADGUN(1)),(RANGER(1),
+ IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
+ (GUNCNTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
+ IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
+ (INFLDR(1),IFINDR(1)),(ROADW(1),ITROWD(1)),(SPRING(1),
+ ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
+ (FUEL(1),IFUEL(1)),(FUELCC(1),IFUELC(1)),(CREW(1),
+ ICREW(1)),(CAPGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
+ (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIR(1),
+ IENVR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
+ (SMoKE(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))

EQUIVALENCE (VEHICLE(1),VFIE(1),IVFIE(1))

CALL TARVAL(IANDV)

CALL OUTCOM

CALL OUTPER(IANDV)
OUTPT ROUTINES - OUTVEH SUBROUTINE

CALL CUTENG(ICSN)

RETURN
FND
SUBROUTINE TABVAL(IV)

ROUTINE TO PUT VALUES OF INTEREST INTO ARRAYS PRIOR TO PRINTING.
ALSO CALCULATES SUMS, AVERAGES, & PERCENTS OF APPROPRIATE ITEMS.

INPUT PARAMETERS:
IV VEHICLE CF INTEREST

COMMON /VFH/
DIMENSION IVEC1E(72,3), IHULL (48,9), ITURET(42,9), FMAING(44,9),
+ FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRENGR(28,7),
+ ISENSR(32,9), ISTALL(30,6), IIGNCT(29,9), IAMMO1(30,10),
+ IAMMOC(26,9), IENGIN(40,8), ITIPS(32,8), IFINDR(28,10),
+ IROA(36,10), ISPRG(30,10), ITRC(34,8), ISKIRT(28,8),
+ IFUEL(28,10), IFUCLC(26,9), ICRA(26,5), IACARGO(26,5),
+ IELECT(26,6), ICOMO1(28,9), IFIREX(28,10), IENVIR(28,9),
+ IDIAGN(26,9), ISIGSP(26,24), ISMOKE(26,10), IESYS(26,10),
+ IRELN(50), VFILE(1), IVFILE(1)

EQUIVALENCE (VEHICLE(1), IVEC1E(1), (HULL(1), IHULL(1), (ITURET(1),
+ (FMACGN(1), FMISGN(1), (IADGUN(1), IADGUN(1), (RENGR(1),
+ (SENSR(1), ISENSR(1), (STALL(1), ISTALL(1),
+ (IGNCT(1), IIGNCT(1), (AMMO1(1), IAMMO1(1), (AMMOC(1),
+ (IAMMOC(1), (ENG1N(1), IENG1N(1), (TRANSM(1), ITRANSM(1),
+ (FINLD1R(1), IFINDP(1), (ROADWH(1), IROADW(1), (SPRING(1),
+ ISPRNG(1), (TRC(1), ITRC(1), (SKIRT(1), ISKIRT(1),
+ (FUEL(1), IFUEL(1), (FUEL(1), IFUEL(1), (CREW(1),
+ (ICRA(1), (ACARO(1), (ELECT(1), IELEC(1),
+ (COMO1(1), ICOMO1(1), (FIREX(1), IFIREX(1), (EVIRO(1),
+ (ENHB(1), (DIAGN(1), IDIAGN(1), (SIGSP(1), SIGSP(1),
+ (SMOKE(1), ISMOKE(1), (ESYS(1), IESYS(1), (RELN(1), IRELN(1))

EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))

COMMON /VPROF/
REAL*8 FLONAM, RECNAM, DFALUT
COMMON /IPAR$/
OUTPUT ROUTINES - TABVAL SUBROUTINE

C
9 COMMON /JPARS1/
10 COMMON /NPARS/
11 COMMON /OUTPLT/
12 LOGICAL*1 OF
13 REAL*8 XNAT,XMANUF,XMODEL
14 COMMON /AUX/
15 REAL*8 BL8,CASES
16 EQUIVALENCE (FNULL,NUL),(IBLS,BLS)
17 COMMON /ARRAYS/
18 REAL*8 MANIF,MODEL,NATION
19 DIMENSION ARRAY(1)
20 EQUIVALENCE (ARRAY(1),DETRNG(1))
21 COMMON /JPARS2/
22 COMMON /DATA/
23 COMMON /CATA2/
C
C INITIALIZE ARRAYS
24 CALL IZERO(NUMBER,NCMPT1*12)
25 CALL INITVL(CF,2*NCMPTP,' ')
26 CALL INITVL(XMANLF,64,BLS)
27 CALL INITVL(XMODEL,64,BLS)
28 CALL INITVL(XNAT,64,BLS)
C
C ITERATE OVER COMPONENT TYPES
29 DO 100 ICMP=1,NCMPTP
C
C GET INFORMATION ABOUT THIS COMPONENT TYPE
30 IRECTP = ICMP + 1
31 IDFF = IFLP(IRECTP)
32 NF = NFLOD(IRECTP)
C
C ITERATE OVER COMPONENTS OF THIS VEHICLE
33 IRFC = IVECLE(ICMP,IV)
34 10 IF (IRFC .EQ. NULL) GO TO 20
C
C INCREMENT NUMBER OF COMPONENTS OF THIS TYPE
OUTPUT ROUTINES - TABVAL SUBROUTINE

35    NUMBER(ICMP) = NUMBER(ICMP) + 1

36    C INCREMENT WEIGHT OF COMPONENTS OF THIS TYPE
    C    WEIGHT(ICMP) = WEIGHT(ICMP) + DVAL(VFILE(IOFF),NF,JWT,IREC,
    +   IRECTP,'LB ',DF(3,ICMP))

37    C INCREMENT INTERNAL VOLUME OF COMPONENTS OF THIS TYPE
    C    ICLOC = IVVAL(VFILE(IOFF),NF,JLOC,IREC)
    38    IF( ICLOC .EQ. 3) GO TO 12
    39    VOLUME(ICMP) = VOLUME(ICMP) + DVAL(VFILE(IOFF),NF,JVOL,IREC,
    +   IRECTP,'FT3 ',DF(4,ICMP))

40    C INCREMENT MAINTENANCE / OPERATIONAL HRS
    12    DWNUP(ICMP) = DWNUP(ICMP) + DVAL(VFILE(IOFF),NF,JDWNUP,IREC,
    +   IRECTP,'-',DF(5,ICMP))

41    C INCREMENT MATURITY INDEX
    41    MATUR(ICMP) = MATUR(ICMP) + DVAL(VFILE(IOFF),NF,JMATUR,IREC,
    +   IRECTP,'-',DF(6,ICMP))

42    C INCREMENT COMPLEXITY INDEX
    42    JCMPLX(ICMP) = JCMPLX(ICMP) + DVAL(VFILE(IOFF),NF,JCMPPLX,IREC,
    +   IRECTP,'-',DF(7,ICMP))

43    C INCREMENT RELIABILITY INDEX
    43    RELIAB(ICMP) = RELIAB(ICMP) + DVAL(VFILE(IOFF),NF,JRELIB,IREC,
    +   IRECTP,'-',DF(8,ICMP))

44    C DO FOLLOWING OPERATIONS ONLY FOR FIRST COMPONENT OF THIS TYPE
    C    IF (NUMBER(ICMP) .NE. 1) GO TO 15

45    C NUMBER OF ITEMS PRODUCED
    C    NPROD(ICMP) = IOVAL(VFILE(IOFF),NF,JNUM,IREC,
OUTPUT ROUTINES - TABVAL SUBROUTINE

C
+ IRECTP,'-',DF(1,ICMP))
C
C R&D TIME
C RDTIME(ICMP) = DVAL(VFILE(IOFF),NF,FRTIME,IREC,
+ IRECTP,'YR',DF(2,ICMP))
C
C NOTE COMPONENT INDEX
C INDX(ICMP) = IVAL(VFILE(IOFF),NF,JID,IREC)
C
C NATION
C K = IVAL(VFILE(IOFF),NF,JNAT,IREC)
C XNAT(ICMP) = NATION(K)
C
C MANUFACTURER
C K = IVAL(VFILE(ICFF),NF,JMANUF,IREC)
C XMANUF(ICMP) = MANUF(K)
C
C MODEL
C K = IVAL(VFILE(ICFF),NF,JMODEL,IREC)
C XMODEL(ICMP) = MODEL(K)
C
C GET NEXT COMPONENT OF THIS TYPE
C IREC = IVAL(VFILE(IOFF),NF,JNEXT,IREC)
C GO TO 10
C
C IF THERE IS MORE THAN ONE COMPONENT OF THIS TYPE, COMPUTE
C AVERAGES FOR CERTAIN OF THE ATTRIBUTES
C
C IF (NUMBER(ICMP) .LE. 1) GO TO 100
C
C X = NUMBER(ICMP)
C
C AVERAGE COMPLEXITY INDEX
C ICMPX(ICMP) = (ICMPX(ICMP) / X) + 0.5
C
C AVERAGE MATURITY INDEX
OUTPUT ROUTINES - TABVAL SUBROUTINE

59 MATUR(ICMP) = (MATUR(ICMP) / X) + 0.5

60 100 CONTINUE

C COMPUTE MEASURES RELEVANT TO VEHICLE AS A WHOLE

61 WEIGHT(NCMPT1) = SUM(WEIGHT,NCMPTP)
62 WEIGHT(NCMPT1) = (1. + FMCWT) * WEIGHT(NCMPT1)
63 VOLUME(NCMPT1) = VOLUME(JHULL) + VOLUME(JTURST)
64 CALL ARMULT(PCWT,WEIGHT,NCMPT1,100./WEIGHT(NCMPT1))
65 CALL ARMULT(PCVCL(3),VOLUME(3),NCMPT1-2,100./VOLUME(NCMPT1))
66 S = SUM(CMPWTS,NCMPTP)
67 MATUR(NCMPT1) = 0.5 + SUMPD2(MATUR,CMPWTS,NCMPTP) / S
68 ICMLX(NCMPT1) = 0.5 + SUMPD2(ICMPLX,CMPWTS,NCMPTP) / S
69 DWNUP(NCMPT1) = SUM(DWNUP,NCMPTP)
70 RELIAB(NCMPT1) = 0.0
71 DO 160 ICM=1,NCMPTP
72 IF(RELIAB(ICM) .EQ. 0.0) GO TO 160
73 RELIAB(NCMPT1) = RELIAB(NCMPT1) + 1./RELIAB(ICM)
74 160 CONTINUE

C IF(RELIAB(NCMPT1) .NE. 0.0) RELIAB(NCMPT1)=1./RELIAB(NCMPT1)

76 VEHICLE(JVWT,IV) = WEIGHT(NCMPT1)
77 VEHICLE(JVVOL,IV) = VOLUME(NCMPT1)
78 RETURN
79 END
C.6: UTILITY ROUTINES

This section contains listings of utility routines used for a variety of purposes. Comment cards in the listing of each routine indicate its purpose.
UTILITY ROUTINES - ARMULT SUBROUTINE

SUBROUTINE ARMULT(Y,X,N,W)
C
C ROUTINE TO MULTIPLY EACH ELEMENT OF A VECTOR BY A CONSTANT,
C STORING THE RESULT OF EACH MULTIPLICATION IN THE CELLS OF A
C VECTOR PARALLEL TO THE FIRST.
C
C INPUT PARAMETERS:
C Y RESULTANT VECTOR
C X INITIAL VECTOR
C N NUMBER OF CELLS IN X & Y
C W THE CONSTANT THAT EACH ELEMENT OF X IS TO MULTIPLIED BY
C
DIMENSION X(N), Y(N)
C
DO 10 I=1,N
10 Y(I) = X(I) * W
RETURN
END
UTILITY ROUTINES - COPY SUBROUTINE

C-1 25

1 SUBROUTINE COPY(X,Y,N)
   C ROUTINE TO COPY VECTOR X TO VECTOR Y.
   C
   C INPUT PARAMETERS:
   C X VECTOR TO BE COPIED
   C Y VECTOR TO RECEIVE CONTENTS OF X
   C N SIZE OF VECTORS X AND Y
   C
   2 DIMENSION X(N),Y(N)
   C
   3 DO 10 I=1,N
   4   10 Y(I) = X(I)
   5 RETURN
   6 END
UTILITY ROUTINES - DVAL FUNCTION

1
FUNCTION DVAL(FILE,NFLD,IFLD,IREC,IRECTP,UNIT,DF)

C FUNCTION TO RETRIEVE A VALUE OF A SPECIFIED FIELD OF A RECORD.
C IF THE VALUE IS NULL A DEFAULT VALUE IS SUBSTITUTED FOR IT. AN
C INPUT TO THE FUNCTION ALSO INDICATES WHAT UNITS OF MEASURE ARE
C DESIRED. IF THE VALUE IS NOT ALREADY IN THESE UNITS IT IS CONVERTED
C TO THEM.
C
C INPUT PARAMETERS:
C FILE     FILE IN WHICH VALUE IS LOCATED
C NFLD     NUMBER OF FIELDS IN EACH RECORD OF FILE
C IFLD     FIELD OF INTEREST
C IREC     RECORD OF INTEREST
C IRECTP   RECORD TYPE
C UNIT     UNIT OF MEASURE DESIRED
C
C OUTPUT PARAMETERS:
C DF       SET TO "*" IF DEFAULT VALUE IS USED
C
2 COMMON / VPROTO/ 
3 REAL*8 FLDNAM,RECNAM,DFault
4 COMMON / NPARS/ 
5 COMMON / AUX/ 
6 REAL*8 BL8,DASHES
7 EQUIVALENCE (FNULL,NULL),(IBLS,BLS)

C LOGICAL*1 STAR,DF
9 DIMENSION FILE(NFLD,1)
10 DATA DASH/'-'/,STAR/**/

C DVAL = FILE(IFLD,IREC)
12 IFLDRC = IFLDPT(IRECTP) + IFLD
13 IF (DVAL .NE. FNULL) GO TO 5
14 DVAL = DFault(IFLDRC)
15 DF = STAR
UTILITY ROUTINES - DVAL FUNCTION

ISN
16 5 IF (UNIT .EQ. DASH .OR. UNIT .EQ. UNITS(IFLDRC) .OR. UNITS(IFLDRC) + .EQ. CASH) RETURN
   C SEARCH TRANSLATION TABLE FOR UNIT TRANSLATION FACTOR
17   DO 10 I=1,NUTRNS
18    IF (UNIT .NE. TUNITS(1,I)) .OR. UNITS(IFLDRC) .NE. TUNITS(2,I)) + GO TO 8
19   DVAL = DVAL * TUNITS(3,I)
20 RETURN
21   GO TO 10
21 8 IF (UNIT .NE. TUNITS(2,I)) .OR. UNITS(IFLDRC) .NE. TUNITS(1,I)) + GO TO 10
22   DVAL = DVAL / TUNITS(3,I)
23 RETURN
24   CONTINUE
   C
   C WRITE ERROR MESSAGE THAT UNIT TRANSLATION NOT FOUND
25   CALL ERR(37, 'DVAL ', IRECTP, UNIT, UNITS(IFLDRC))
26 RETURN
27 END
UTILITY ROUTINES - DVAL2 FUNCTION

FUNCTION DVAL2(X, ICMP, IFLD, UNIT)

FUNCTION TO RETRIEVE A VALUE OF A SPECIFIED FIELD OF A RECORD.
IF THE VALUE IS NULL A DEFAULT VALUE IS SUBSTITUTED FOR IT. AN
INPUT TO THE FUNCTION ALSO INDICATES WHAT UNITS OF MEASURE ARE
DESIZED. IF THE VALUE IS NOT ALREADY IN THESE UNITS IT IS CONVERTED
TO THEM.

INPUT PARAMETERS:
X VALUE OF FIELD
ICMP COMPONENT TYPE OF RECORD
IFLD FIELD OF RECORD OF INTEREST
UNIT UNIT OF MEASURE DESIRED

COMMON /VPROTO/
REAL*8 FLDNAM, RECNAM, DEFAULT
COMMON /NPARS/
COMMON /AUX/
REAL*8 BL8, DASHES
EQUIVALENCE (FNULL, NULL), (IBLS, BLS)

DATA DASH /*-*/

DVAL2 = X
IFLORDC = IFLCPT(ICMP+1) + IFLD
IF (DVAL2 .EQ. FNULL) DVAL2 = DEFAULT(IFLORDC)
IF (UNIT .EQ. CASH .OR. UNIT .EQ. UNITS(IFLORDC) .OR. UNITS(IFLORDC)
+ .EQ. DASH) RETURN
SEARCH TRANSLATION TABLE FOR UNIT TRANSATION FACTOR
DO 10 I = 1, NUTRNS
  IF (UNIT .NE. TUNITS(1, I) .OR. UNITS(IFLORDC) .NE. TUNITS(2, I))
    GO TO 8
  DVAL2 = DVAL2 * TUNITS(3, I)
  RETURN
8 IF (UNIT .NE. TUNITS(2, I) .OR. UNITS(IFLORDC) .NE. TUNITS(1, I))
UTILITY ROUTINES - DVAL2 FUNCTION

ISN

+    GO TO 10
18     DVAL2 = DVAL2 / UNITS(3,I)
19     RETURN
20    10     CONTINUE
C
C WRITE ERROR MESSAGE THAT UNIT TRANSLATION NOT FOUND
21     CALL ERR(37,'DVAL2 ',ICMP+1,UNIT,UNITS(IFLORC))
22     RETURN
23     END
UTILITY ROUTINES — ERR SUBROUTINE

SUBROUTINE ERR(IERR, SUBR, PAR1, PAR2, PAR3)

ROUTINE TO PROCESS ERRORS AND WRITE OUT ERROR COMMENTS.
ALSO HAS AN ENTRY "WARN" TO PRINT OUT WARNING MESSAGES.

INPUT PARAMETERS:
IERR   POSITIVE INTEGER IDENTIFYING THE ERROR CONDITION
SUBR   SUBROUTINE IN WHICH THE ERROR OCCURRED
PAR1   THE FIRST PARAMETER OF THE ERROR CONDITION
PAR2   THE SECOND PARAMETER OF THE ERROR CONDITION
PAR3   THE THIRD PARAMETER OF THE ERROR CONDITION

COMMON /VFH/
DIMENSION IVECLE(72, 3), I Hull(48, 9), ITURRET(42, 9), FMAING(44, 9),
+ FMAGN(40, 10), FMISGN(40, 7), IADGUN(40, 6), IIRANGR(28, 7),
+ ISENS(32, 9), ISTABL(30, 6), IGNCTL(28, 9), IAMMO(30, 30),
+ IAPPLE(26, 26), IENGIN(40, 8), ITRANS(32, 8), IFINDR(28, 10),
+ IMOADW(26, 10), ISPRNG(30, 10), ITRACK(34, 8), ISKIRT(28, 8),
+ IFUEL(28, 10), IFUELC(26, 9), ICREW(26, 5), ICAPGO(26, 5),
+ IELECT(26, 6), ICMMD(28, 9), IFIREX(28, 11), IEVIR(26, 9),
+ IDIAGN(26, 5), ISIGSP(26, 24), ISMOKE(26, 10), IRELSYS(26, 10),
+ IRELN(3, 50), VFILE(1), VFILE(1)

EQUIVALENCE (VEHICLE(1), IVECLE(1)), (I HULL(1), I HULL(1)), (ITURRET(1),
+ ITURRET(1)), (IADGUN(1), FMAING(1)), (MACGUN(1), FMAGN(1)),
+ (FMISGN(1), FMISGN(1)), (IADGUN(1), IADGUN(1)), (I RANGR(1),
+ IRANGR(1)), (ISENS(1), ISENS(1)), (ISTABL(1), ISTABL(1)),
+ (IGNCTL(1), IGNCTL(1)), (IAMMO(1), IAMMO(1)), (IAMMO(1),
+ IAMMO(1)), (IENGIN(1), IENGIN(1)), (ITRANS(1), ITRANS(1)),
+ (IFINDR(1), IFINDR(1)), (IMOADW(1), IMOADW(1)), (ISPRNG(1),
+ ISPRNG(1)), (ITRACK(1), ITRACK(1)), (ISKIRT(1), ISKIRT(1)),
+ (IFUEL(1), IFUEL(1)), (IFUELC(1), IFUELC(1)), (ICREW(1),
+ ICREW(1)), (ICAPGO(1), ICAPGO(1)), (IELECTR(1), IELECTR(1)),
+ (ICOMMC(1), ICOMMC(1)), (IFRX(1), IFRX(1)), (IEVIR(1),
+ IEVIR(1)), (IDIAIGN(1), IDIAGN(1)), (ISIGSP(1), ISIGSP(1)),
+ (ISMOKEG(1), ISMOKE(1)), (I EWSYS(1), I EWSYS(1)), (IREL(1), IREL(1))
UTILITY ROUTINES - ERR SUBROUTINE

ISN

5 EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))
6 COMMON /VPROTO/
7 REAL*8 FLDSAM,RECNAME,DEFAULT
8 COMMON /JPARS/
9 COMMON /JPARS1/
10 COMMON /NPARS/
11 COMMON /OUTPLT/
12 LOGICAL*1 DF
13 REAL*8 XNAT,XMANUF,XMODEL
14 COMMON /AUX/
15 REAL*8 BLR,CASHES
16 EQUIVALENCE (FNULL, NULL),(IBLS, BLS)
17 COMMON /ARRAYS/
18 REAL*8 MANUF,MODEL,NATION
19 DIMENSION ARRAY(1)
20 EQUIVALENCE (ARRAY(1),DETRNG(1))
21 COMMON /JPARS2/
22 COMMON /SPECS/
23 REAL*8 ATEENG,ATRPER,VEHCAT,VEHCM,VEHENG,VEHPER
24 COMMON /SPECS2/
25 REAL*8 VEH
26 COMMON /CATA/
27 COMMON /CATA2/

C
28 REAL*8 SUBR
29 DIMENSION PAR1(2),PAR2(2)
30 EQUIVALENCE (XPAR1,IPAR1),(XPAR2,IPAR2),(XPAR3,IPAR3)

C EQUATE ARCS SO THAT INTEGER PARAMETERS ARE DEFINED.

C
31 XPAR1 = PAR1(1)
32 XPAR2 = PAR2(1)
33 XPAR3 = PAR3

C
34 WRITE (6,2) IERR, SUBR
35 FORMAT ('0 *** ERROR ',I2,' DETECTED IN SUBROUTINE ','A6',')
UTILITY ROUTINES - ERR SUBROUTINE

GO TO (10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320, 330, 340, 350, 360, 370, 380, 390, 400, 410, 420, 430, 440, 450), IERR

37 10  WRITE (6, 15) XPAR1, XPAR2
38   15  FORMAT (' THE FIRST NON-COMMENT RECORD OF THE PROTOTYPE FILE /
+ ' WAS NOT A VALID VEHICLE DEFINITION RECORD. IT CONTAINED',
+ '*A4/'IN COLUMNS 2-5, AND ', A4, ' IN COLUMNS 12-15. ')
39   STOP 1

40  20  WRITE (6, 25) PARI, IPAR3, IPAR2
41   25  FORMAT (' THE ATTRIBUTE ', 2'A4, ' OF RECORD TYPE ', 12, ' SPECIFIED /
+ ' IN THE VEHICLE PROTOTYPE FILE HAS AN OFFSET WITHIN ITS',
+ ' RECORD OF ', 14, ' WHICH EXCEEDS THE NUMBER OF FIELDS /
+ ' DEFINED FOR THIS RECORD TYPE. ')
42  PAUSE
43  GO TO 9999

44  30  WRITE (6, 35) IPAR2, IPAR3, PARI
45   35  FORMAT (' THERE WERE ', 13, ' RECORD TYPES ENCOUNTERED IN THE',
+ ' VEHICLE PROTOTYPE FILE AND ONLY ', 13, ' DEFINED.' /
+ ' THE LATEST RECORD READ CONTAINS ', 2'A4, ' IN COLS 2-9. ')
46  PAUSE
47  GO TO 9999

48  40  WRITE (6, 45) PARI, PAR2
49   45  FORMAT (' THE ALTERNATIVE COMPONENT FILE SPECIFIED THE VEHICLE /
+ ' CATEGORY ', 2'A4, ', WHICH DOES NOT MATCH THE VEHICLE CATEGORY/
+ ' SPECIFIED BY THE PROTOTYPE FILE: ', 2'A4)
50  PAUSE
51  GO TO 9999

52  50  WRITE (6, 55) PARI, IPAR3
53   55  FORMAT (' A HEADER WAS ENCOUNTERED IN THE ALTERNATIVE COMPONENT')
UTILITY ROUTINES - ERR SUBROUTINE

      +  ' FILE IN WHICH THE COMPONENTS WERE NOT SEQUENTIALLY' /
      +  ' NUMBERED, THE OFFENDING COMPONENT TYPE IS ',2A4,' THE' /
      +  ' OFFENDING COMPONENT SEQUENCE NUMBER IS ',I3) /

54   PAUSE
55   GO TO 9999

C
56  60   WRITE (6,65) PAR2,PARI
57  65   FORMAT (' THE ATTRIBUTE ',2A4,' APPEARING UNDER COMPONENT ',2A4/
      +  ' IN THE ALTERNATIVE VEHICLE FILE COULD NOT BE FOUND IN THE' /
      +  ' LIST OF LEGAL ATTRIBUTES FOR THIS COMPONENT.' )
58   PAUSE
59   GO TO 9999

C
60  70   WRITE (6,75) PARI
61  75   FORMAT (' THE RECORD TYPE SPECIFIED, ',I3,' WAS OUT OF BOUNDS.' )
62   PAUSE
63   GO TO 9999

C
64  80   WRITE (6,85) IPARI,IPAR2
65  85   FORMAT (' HAVE RUN OUT OF RECORDS OF TYPE ',I3,'.' /
      +  ' MORE THAT ',I4,' RECORDS OF THIS TYPE WERE NEEDED.' /
      +  ' NEED TO REDIMENSION APPROPRIATE ARRAYS, CHANGE BLOCKDATA,' /
      +  ' AND RECOMPILE.' )
66   PAUSE
67   GO TO 9999

C
68  90   WRITE (6,95) PARI
69  95   FORMAT (' THE HEADER ',2A4,' READ FROM THE ALTERNATIVE COMPONENT' /
      +  ' FILE WAS NOT RECOGNIZED AS A LEGAL COMPONENT.' )
70   PAUSE
71   GO TO 9999

C
72 100  WRITE (6,105) XPAR3,PAR2,PARI
73 105  FORMAT (' THE DATATYPE ',A4,' STORED FOR ATTRIBUTE ',2A4,' OF' /
      +  ' COMPONENT ',2A4,' DOES NOT MATCH ANY OF THE LEGAL' /
      +  ' ALTERNATIVES.' )
UTILITY ROUTINES - ERR SUBROUTINE

15 SN
74     PAUSE
75     GO TO 9999
76 C
77 11) WRITE (6,115) XPAR1,XPAR2
78 115 FORMAT (' THE IDENTIFYING HEADER ',A4,' READ FROM THE USER '/
79 +    ' SPECIFICATION FILE DID NOT MATCH THAT EXPECTED: ',A4)
80 PAUSE
81 GO TO 9999
82 C
83 12) WRITE (6,125) PAR1,PAR2
84 125 FORMAT (' THE USER SPECIFICATION FILE DESIGNATED A VEHICLE'/
85 +    ' CATEGORY ',2A4,' WHICH DOES NOT MATCH THE VEHICLE CATEGORY'/
86 +    ' SPECIFIED BY THE PROTOTYPE FILE: ',2A4)
87 PAUSE
88 GO TO 9999
89 C
90 130 WRITE (6,135) PAR1,IPAR2
91 135 FORMAT (' AN INSUFFICIENT NUMBER OF ',2A4,' RECORDS, ',13,'/'
92 +    ' WERE SET ASIDE TO STORE THE USERS SPECS. ARRAYS'/
93 +    ' DIMENSIONED ON THIS SUBSCRIPT SHOULD BE ENLARGED, BLOCK'/
94 +    ' DATA CHANGED, AND THE PROGRAM RECOMPILED. ')
95 PAUSE
96 GO TO 9999
97 C
98 140 WRITE (6,145) XPAR1
99 145 FORMAT (' THE FIRST CHARACTER OF A HEADER RECORD FOR '/
100 +    ' (1) COMPONENTS, (2) ENGINEERING PARAMETERS, OR (3) (VEHICLE)'/
101 +    ' PERFORMANCE PARAMETERS WAS ''A1'',' NOT ''C'', ''E'', '
102 +    ''P'', OR ''V'', ')
103 PAUSE
104 GO TO 9999
105 C
106 15) WRITE (6,155) XPAR1
107 155 FORMAT (' THE TYPE FIELD OF A RECORD DESCRIBING A '/
108 +    ' VEHICLE PERFORMANCE PARAMETER CONSTRAINT CONTAINED'/
109 +    ''A3'',' NOT '' '' OR ''VEH'', ')}
UTILITY ROUTINES - ERR SUBROUTINE

ISN
94   PAUSE
95   GO TO 9999

C
96   160   WRITE (6,165) IPAR2,IPAR1
97   165   FORMAT (' THE PROTOTYPE FILE SPECIFIED THAT FIELD ',I3/
+   ' OF RECORD TYPE ',I3,' HAD A DATATYPE "RA"/
+   ' THIS DOES NOT MATCH WITH THE ARRAY IARDF? STORING /
+   ' INFORMATION ABOUT FIELDS WHICH MAY HAVE VALUES/
+   ' WHICH ARE ARRAYS.')
98   PAUSE
99   GO TO 9999

C
100  170   WRITE (6,175) IPAR3,PAR1,IPAR2
101  175   FORMAT (' HAVE RUN OUT OF SPACE IN STORING ARRAY TYPE ',I2/
+   ' AS A VALUE OF ATTRIBUTE ',2A4,'. NEED TO ENLARGE/
+   ' NUMBER OF SUCH ARRAYS BEYOND THE ',I4, ' CURRENTLY/
+   ' ALLOCATED. CHANGE AFFECTED ARRAY SIZES & BLOCKDATA/
+   ' AND RECOMPILE.')
102   PAUSE
103   GO TO 9999

C
104  180   WRITE (6,185) PAR2,PAR1
105  185   FORMAT (' THE USER SPECIFICATION FILE DESIGNATED A VEHICLE /
+   ' CATEGORY ',2A4,' WHICH DOES NOT MATCH THE VEHICLE CATEGORY/
+   ' SPECIFIED BY THE EXISTING VEHICLE FILE: ',2A4)
106   PAUSE
107   GO TO 9999

C
108  190   WRITE (6,195) IPAR1
109  195   FORMAT (' USER COMPONENT SPECIFICATION # ',I3,' CALLED FOR THE /
+   ' USE OF AN EXISTING VEHICLE BY HAVING "VEH" IN COLUMNS 2-4/
+   ' BUT FAILED TO INDICATE WHICH EXISTING VEHICLE.')
110   PAUSE
111   GO TO 9999

C
112  200   WRITE (6,205) PAR1,IPAR2
UTILITY ROUTINES - ERR SUBROUTINE

   ISN  FORMAT ('VEHICLE NAME ',2A4,' DID NOT FIT IN THE VEHs ARRAY.')/
113  205   MUST EXPAND THE SIZE FOR THIS ARRAY BEYOND ',I3')
+                  PAUSE
114  GO TO 9999
+                  WRITE (6,215) IPAR1,IPAR2
116  210   FORMAT ('ARRAY OVERFLOW FOR ARRAY IRECS. MUST EXPAND ITS SIZE'/
+                  'BEYOND ',I3,'. THE VEHICLE NAME BEING PROCESSED WHEN'/
+                  'OVERFLOW OCCURRED WAS ',I2)
118  PAUSE
119  GO TO 9999
+                  WRITE (6,225) XPAR1,PAR2
120  222   FORMAT ('THE COMPONENT CODE ''A2'' SPECIFIED BY THE '/
+                  'ENGINEERING PARAMETER SPECIFICATION REFERRING TO ATTRIBUTE',
+                  'CANNOT BE LOCATED IN THE LIST OF VALID COMPONENT ',
+                  ' CODES. THIS SPECIFICATION WILL BE SKIPPED.')
122  RETURN
+                  WRITE (6,235) IPAR3,IPAR2,IPAR1
123  230   FORMAT ('THE NUMBER OF PARAMETERS CALLED FOR BY JUSER '/
+                  'SPECIFICATION # ',I3,' FOR A COMPONENT OF TYPE ',I2/
+                  'WAS ',I3,' WHICH IS OUTSIDE THE RANGE DEFINED BY THE'/
+                  'MINIMUM AND MAXIMUM FOR THIS COMPONENT TYPE SPECIFIED '/
+                  'IN THE PROTOTYPE FILE. HOWEVER THE USER SPEC WILL BE'/
+                  'USED ANYWAY.')
125  RETURN
+                  WRITE (6,245) IPAR3,IPAR2,IPAR1
126  240   FORMAT ('USER SPECIFICATION # ',I3,' FOR COMPONENT OF TYPE ',A3/
+                  'REFERRED TO COMPONENT INDEX ',I3,' WHICH IS OUT OF BOUNDS'/
+                  'OF THE LEGAL RANGE OF SUCH INDEICES.')
129  PAUSE
129  GO TO 9999
+                  WRITE (6,255) PAR2,IPAR3,XPAR1
130  250
UTILITY ROUTINES - ERR SUBROUTINE

131 255 FORMAT (' THE ATTRIBUTE ',2A4,' SPECIFIED IN THE USER ENGINEERING'/
+ ' PARAMETER CONSTRAINT # ',I3,', COULD NOT BE FOUND AMONGST'/
+ ' THE LEGAL LIST OF ATTRIBUTES FOR COMPONENT ',A2,', ' THIS'/
+ ' SPECIFICATION WILL BE IGNORED."
132 RETURN
133 260 WRITE (6,265) XPAR3,XPAR1,PAR2
134 265 FORMAT (' THE RELOP ''',A4,''' SPECIFIED BY THE USER '/
+ ' SPECIFICATION REFERRING TO COMPONENT ',A2,' AND ATTRIBUTE ',
+ ' A4/' WAS NOT AMONGST THE LIST OF LEGAL RELATIONAL'/
+ ' OPERATORS. THIS SPECIFICATION WILL BE IGNORED."
135 RETURN
136 270 WRITE (6,275) PAR2,IPAR3,RELNAM(IPAR1)
137 275 FORMAT (' A CONSTRAINT ASSOCIATED WITH ATTRIBUTE ',2A4/
+ ' OF RECORD TYPE ',I2,' SPECIFIED A ''',A2,''' RELATION'/
+ ' INVOLVING CHARACTER DATA."
138 PAUSE
139 GO TO 9999
140 280 WRITE (6,285) PAR1
141 285 FORMAT (' THE COMPONENT NAME ',2A4,' SPECIFIED IN THE EXISTING'/
+ ' VEHICLE FILE DOES NOT MATCH ANY OF THE LEGAL COMPONENT'/
+ ' NAMES STORED."
142 PAUSE
143 GO TO 9999
144 290 WRITE (6,295) IPAR2,IPAR3,IPAR1
145 295 FORMAT (' A CONSTRAINT IN THE USER SPECIFICATION FILE INDICATED '/
+ ' THAT FIELD ',I3,' OF RECORD ID ',I3,' HAVING RECORD TYPE ',
+ ' I4,' WAS TO BE NOT EQUAL A SPECIFIED VALUE. THIS IS NOT'/
+ ' A LEGAL CONSTRAINT TO SPECIFY IN THIS SITUATION."
146 PAUSE
147 GO TO 9999
148 300 WRITE (6,305) IPAR1
UTILITY ROUTINES - ERR SUBROUTINE

149 305 FORMAT (' HAVE RUN OUT OF RELATION RECORDS. MUST INCREASE THE'/
+ ' NUMBER ABOVE ',I3,' AND RECOMPILE.')
150 PAUSE
151 GO TO 9999
152 310 WRITE (6,315) IPARI
153 315 FORMAT (' HAVE RUN OUT OF RECORDS IN "OUTNAM". THE LAST ONE'/
+ ' USED WAS ',I4,'. NEED TO INCREASE THE SIZE OF THIS ARRAY.')
154 PAUSE
155 GO TO 9999
156 320 WRITE (6,325)
157 325 FORMAT (' DATA FOR THE FOLLOWING USER SPECIFICATIONS WERE NOT '/
+ ' FOUND IN THE EXISTING VEHICLE FILE: ///','/'
+ ' TYPE SPEC SPEC VEHICLE')
158 IR = IPARI
159 322 IF (IP .EQ. NULL) GO TO 328
160 IV = IRECS(JVEH,IR)
161 WRITE (6,326) IRECS(JTYP,IR),IRECS(JSPEC,IR),VEHS(IV)
162 326 FORMAT (I6,18,6X,A8)
163 IR = IRECS(JXT,IR)
164 GO TO 322
165 328 PAUSE
166 GO TO 9999
167 330 WRITE (6,335) IPARI
168 335 FORMAT (' HAVE ENCOUNTERED A PREMATURE END TO THE EXISTING'/
+ ' VEHICLE FILE. LSTVEH = ',I3)
169 PAUSE
170 GO TO 9999
171 340 CONTINUE
172 350 CONTINUE
173 360 CONTINUE
UTILITY ROUTINES - ERR SUBROUTINE

C
174 370 WRITE (6,375) IPAR1,XPAR3,XPAR2
175 375 FORMAT (' AN ATTEMPT TO TRANSLATE THE VALUE IN A FIELD OF A/
176         ' RECORD OF RECORD TYPE ',I3,' FROM ',A4,' TO ',A4,' FAILED/
177         ' BECAUSE THIS PAIR OF UNITS WAS NOT FOUND IN "TUNITS"."
178     PAUSE
179    GO TO 9999
C
180 380 WRITE (6,385) IPAR1
181 385 FORMAT (' AN OUTPUT ARRAY WAS ENCOUNTERED WITH ',I3,' DIMENSIONS/
182         ' BUT THE PROGRAM WAS WRITTEN TO HANDLE ONLY ARRAYS HAVING/
183         ' 1 TO 3 DIMENSIONS."
184     PAUSE
185    GO TO 9999
C
186 390 CONTINUE
C
187 400 WRITE (6,405) XPAR3,XPAR1,PAR2,PAR2
188 405 FORMAT (' THE DISTANCE ('',F8.1,'') SPECIFIED BY VARIABLE ',A4/
189         ' ASSOCIATED WITH THE SELECTED ROAD WHEEL COMPONENT AND/
190         ' GIVING THE VERTICAL DISTANCE BETWEEN THE ROAD WHEELS AND '/
191         ' A8,' , WAS TOO SMALL TO KEEP THE ROAD WHEELS FROM INTERFERING/
192         ' WITH THE ',A8)
193     PAUSE
194    GO TO 9999
C
195 410 WRITE (6,415) IPAR1,MINNUM,MAXNUM
196 415 FORMAT (' THE NUMBER OF ROAD WHEELS SPECIFIED, ',I2,' WAS NOT/
197         ' BETWEEN THE ALLOWED MINIMUM, ',I2,' AND MAXIMUM, ',I3)
198     PAUSE
199    GO TO 9999
C
200 420 WRITE (6,425) XPAR3,IPAR1
201 425 FORMAT (' A SATISFACTORY CANDIDATE VEHICLE COULD NOT BE FOUND/
202         ' BECAUSE NONE OF THE ',I2,' ALTERNATIVES OF COMPONENT TYPE/
203         ' I3,' WERE FOUND TO BE SUITABLE."
204     PAUSE
205    GO TO 9999
UTILITY ROUTINES - ERR SUBROUTINE

193  PAUSE
194  GO TO 9999

195   430  WRITE (6,435)
196   435  FORMAT (' NONE OF THE COMBINATIONS OF ALTERNATIVE COMPONENTS'/
+ ' WERE BOTH CONSISTENT WITH EACH OTHER AND WITH THE '/
+ ' USERS SPECIFICATIONS. THEREFORE A CANDIDATE VEHICLE'/
+ ' SOLUTION WAS NOT FOUND. THE USER SHOULD RUN THE PROGRAM'/
+ ' AGAIN, REMOVING SOME CONSTRAINTS."

197  PAUSE
198  GO TO 9999

199   440  CONTINUE

200   450  I = IPAR1 - 1
201   455  WRITE (6,455) I

202   455  FORMAT (' THE STABILIZATION PERFORMANCE INDEX SPECIFIED IN THE '/
+ ' SELECTED "STABLE" COMPONENT RECORD WAS OUT-OF-BOUNDS (NOT'/
+ ' 1 OR 2). IT WAS: ',I3)

203  PAUSE
204  GO TO 9999

205  ENTRY WARN(IWARN, SUBR, PAR1, PAR2, PAR3)

206  C  EQUIATE ARGS SO THAT INTEGER PARAMETERS ARE DEFINED
207  C
208  XPAR1 = PAR1(1)
209  XPAR2 = PAR2(1)
210  XPAR3 = PAR3

211  C  BRANCH TO APPROPRIATE PIECE OF CODE TO HANDLE WARNING MESSAGE
212  C
213  GO TO (510,520), IWARN

214  510  ICMP = IPAR1 - 1
215  515  WRITE (6,515) IPAR2, IPAR3, ICMP
UTILITY ROUTINES - ERR SUBROUTINE

212 515 FORMAT ("A CONSTRAINT SPECIFIED IN THE USER SPECIFICATION FILE/
+ ' HAS BEEN USED TO CHANGE THE VALUE OF FIELD ',I3,' OF '/
+ ' RECORD ',I3,' OF COMPONENT TYPE ',I3,'. IF THIS'/
+ ' MODIFICATION OF THE COMPONENT REQUIRES DEVELOPMENT TIME'/
+ ' OR COST, OR IF THE COMPONENT HAS BEEN USED TO DEPLOY'/
+ ' PERFORMANCE DATA FOR THIS RUN, ONE SHOULD TAKE THESE'/
+ ' FACTORS INTO ACCOUNT IN INTERPRETING THE OUTPUT OF THIS',
+ ' RUN."
)
213 RETURN

214 520 WRITE (6,525)
215 525 FORMAT ("THE GUN DEPRESSION ANGLE SPECIFIED IN THE SELECTED/
+ ' MAIN GUN COULD NOT BE ACHIEVED WITH THE HULL & TURRET'/
+ ' DIMENSIONS GIVEN. THE MAIN GUNS HEIGHT ABOVE THE ',A8/
+ ' OF THE HULL WAS THEREFORE INCREASED FROM ',F6.0,' INCHES'/
+ ' TO ',F6.0,' INCHES."
)
216 RETURN

217 9999 RETURN
218 END
UTILITY ROUTINES - FVAL1 FUNCTION

ISN
1

FUNCTION FVAL1(X,R,N,V)
C
C FUNCTION TO INTERPOLATE OR EXTRAPOLATE ON AN ARRAY AND RETRIEVE THE
C FUNCTIONAL VALUE CORRESPONDING TO AN INDEPENDENT VARIABLE. THIS
C VALUE IS RETURNED BY THE FUNCTION. THERE ARE TWO OTHER ENTRY POINTS,
C FVAL2 & FVAL3, FOR FUNCTIONS OF TWO OR THREE VARIABLES. THIS
C FUNCTION USES A ROUGH APPROXIMATION METHOD OF INTERPOLATION IN THESE
C LATTER CASES.
C
C INPUT PARAMETERS:
C    X      ARRAY STORING THE FUNCTION
C    R      VECTOR STORING VALUES OF DEPENDENT VARIABLES
C    N      NUMBER OF VALUES TABULATED IN R & N
C    V      INDEPENDENT VALUE TO USE IN INTERPOLATION
C
2
DIMENSION X(N),R(N)
3
DIMENSION XX(N1,N2),R1(N1),R2(N2)
4
DIMENSION XXX(N1,N2,N3),R3(N3)
5
6
DO 10 I=2,N
     IF (V.GT. R(I) .AND. I .NE. N) GO TO 10
     SLOPE = (X(I) - X(I-1)) / (R(I) - R(I-1))
     FVAL1 = X(I-1) + (V - R(I-1))*SLOPE
     RETURN
10 CONTINUE
11 PAUSE
C
12 ENTRY FVAL2(XX,R1,N1,V1,R2,N2,V2)
C
13 DO 30 I=2,N1
     IF (V1.GT. R1(I) .AND. I .NE. N1) GO TO 31
14     DO 20 J=2,N2
15     IF (V2.GT. R2(J) .AND. J .NE. N2) GO TO 21
16     SLOPE1 = (XX(I,J) - XX(I-1,J)) / (R1(I) - R1(I-1))
17     SLOPE2 = (XX(I-1,J) - XX(I-1,J-1)) / (R2(J) - R2(J-1))
18     RETURN
20 CONTINUE
30 CONTINUE
31 PAUSE
C
UTILITY ROUTINES - FVAL1 FUNCTION

ISN
19
FVAL2 = XX(I-1,J-1) + (V1 - R1(I-1))*SLOPE1
       + (V2 - R2(J-1)) * SLOPE2
20
    RETURN
21
20
    CONTINUE
22
30
    CONTINUE
23
    PAUSE
24
    ENTRY FVAL3(XXX,R1,N1,V1,R2,N2,V2,R3,N3,V3)
25
    DO 60 I=2,N1
26
       IF (V1 .GT. R1(I) .AND. I .NE. N1) GO TO 60
27
       DO 50 J=2,N2
28
       IF (V2 .GT. R2(J) .AND. J .NE. N2) GO TO 50
29
       DO 40 K=2,N3
30
       IF (V3 .GT. R3(K) .AND. K .NE. N3) GO TO 40
31
       SLOPE1 = (XXX(I,J,K) - XXX(I-1,J,K)) / (R1(I) - R1(I-1))
32
       SLOPE2 = (XXX(I-1,J,K) - XXX(I-1,J-1,K)) / (R2(J) - R2(J-1))
33
       SLOPE3 = (XXX(I-1,J-1,K) - XXX(I-1,J-1,K-1)) /
                (R3(K) - R3(K-1))
34
       FVAL3 = XXX(I-1,J-1,K-1) + (V1 - R1(I-1))*SLOPE1
       + (V2 - R2(J-1)) * SLOPE2
       + (V3 - R3(K-1)) * SLOPE3
35
    RETURN
36
40
    CONTINUE
37
50
    CONTINUE
38
60
    CONTINUE
39
    PAUSE
40
    END
UTILITY ROUTINES - IATTR FUNCTION

FUNCTION IATTR(ATTR,CMP,IRECTP)

FUNCTION FOR COMPUTING THE INDEX OF AN ATTRIBUTE WITHIN A
COMPONENT RECORD. FUNCTION RETURNS THIS INDEX AS ITS VALUE OR
RETURNS A NULL VALUE IF THE SPECIFIED ATTRIBUTE NAME DOES NOT
IDENTIFY A LEGAL ATTRIBUTE OF THIS COMPONENT. FUNCTION ALSO
RETURNS THE INDEX OF THE RECORD TYPE CORRESPONDING TO CMP.

INPUT PARAMETERS:
ATTR  ATTRIBUTE NAME
CMP   COMPONENT NAME

OUTPUT PARAMETERS:
IRECTP RECORD TYPE INDEX OF CMP

ALSO THERE IS AN ENTRY POINT IATTR2 WHERE BOTH ATTR AND IRECTP ARE
INPUT AND THE FIELD INDEX COMPUTED.

COMMON /VPRCTO/
REAL*8 FLONAM,RECNAM,DFAULT
COMMON /NPARS/
COMMON /AUX/
REAL*8 BLR,DASHFS
EQUIVALENCE (FNULL,NULL),(IBLS,BLS)

REAL*8 ATTR,CMP

DETERMINE INDEX OF RECORD STORING COMPONENT
DO 10 IRECTP=1,NRECTP
    IF (RECNAM(IRECTP) .EQ. CMP) GO TO 15
  10    CONTINUE
    CALL ERR(6,'IATTR ',CMP,ATTR,0)
    IATTP = NULL
    IRECTP = NULL
  15  RETURN
UTILITY ROUTINES - IATTR FUNCTION

C
16 ENTRY IATTR2(ATTR,IRECTP)
C
C DETERMINE INDEX OF ATTRIBUTE WITHIN COMPONENT RECORD
17 15 J1 = IFLOPT(IRECTP) + 1
18 JN = IFLOPT(IRECTP+1)
19 DO 20 J=J1,JN
20 IF (ATTR .EQ. FLDNAM(J)) GO TO 25
21 20 CONTINUE
22 IATTR = NULL
23 IATTR2 = IATTR
24 RETURN
C
25 25 IATTR = IFLOVL(J)
26 IATTR2 = IATTR
27 RETURN
28 END
UTILITY ROUTINES - IDVAL FUNCTION

1

FUNCTION IDVAL(IFILE,NFLD,IFLD,IREG,IRECTP,UNIT,DF)

C FUNCTION TO RETRIEVE A VALUE OF A SPECIFIED FIELD OF A RECORD.
C IF THE VALUE IS NULL A DEFAULT VALUE IS SUBSTITUTED FOR IT. AN
C INPUT TO THE FUNCTION ALSO INDICATES WHAT UNITS OF MEASURE ARE
C DESIRED. IF THE VALUE IS NOT ALREADY IN THESE UNITS IT IS CONVERTED
C TO THEM.

C INPUT PARAMETERS:
C IFILE  FILE IN WHICH VALUE IS LOCATED
C NFLD   NUMBER OF FIELDS IN EACH RECORD OF FILE
C IFLD   FIELD OF INTEREST
C IREG   RECORD OF INTEREST
C IRECTP RECORD TYPE
C UNIT   UNIT OF MEASURE DESIRED

C OUTPUT PARAMETERS:
C DF     SET TO "+" IF DEFAULT VALUE IS USED

COMMON /VPROTO/
REAL*8 FLDNAM*,RECNAM,DFault
COMMON /NPARS/
COMMON /AUX/
REAL*8 BLB,DASHES
EQUIVALENCE (FNULL,NULL),(IBLS,BLS)

LOGICAL*1 STAR,DF
DIMENSION IFILE(NFLD,1)
DATA CASH/"-",STAR/

IDVAL = IFILE(IFLD,IREG)
IFLDC = IFLDPT(IRECTP) + IFLD
IF (IDVAL .NE. NULL) GO TO 5
IDVAL = DFault(IFLDC)
DF = STAR
UTILITY ROUTINES - IDVAL FUNCTION

16 5 IF (UNIT .EQ. CASH .OR. UNIT .EQ. UNITS(IFLDRC) .OR. UNITS(IFLDRC) + .EQ. DASH) RETURN
   C SEARCH TRANSLATION TABLE FOR UNIT TRANSLATION FACTOR
17   DO 10 I=1,NUTRNS
18   IF (UNIT .NE. TUNITS(1,I) .OR. UNITS(IFLDRC) .NE. TUNITS(2,I)) + GO TO 8
19   IDVAL = IDVAL * TUNITS(3,I)
20   RETURN
21  8 IF (UNIT .NE. TUNITS(2,I) .OR. UNITS(IFLDRC) .NE. TUNITS(1,I)) + GO TO 10
22   IDVAL = IDVAL / TUNITS(3,I)
23   RETURN
24 10 CONTINUE
   C WRITE ERROR MESSAGE THAT UNIT TRANSLATION NOT FOUND
25   CALL ERR(37,'IDVAL ',IRECTP,UNIT,UNITS(IFLDRC))
26   RETURN
27   END
UTILITY ROUTINES - IDVAL2 FUNCTION

ISY
1 FUNCTION IDVAL2(Ix, ICMP, IFLD, UNIT)

C FUNCTION TO RETRIEVE A VALUE OF A SPECIFIED FIELD OF A RECORD.
C IF THE VALUE IS NULL A DEFAULT VALUE IS SUBSTITUTED FOR IT. AN
C INPUT TO THE FUNCTION ALSO INDICATES WHAT UNITS OF MEASURE ARE
C DESIRED. IF THE VALUE IS NOT ALREADY IN THESE UNITS IT IS CONVERTED
C TO THEM.

C INPUT PARAMETERS:
C X VALUE OF FIELD
C ICMP COMPONENT TYPE OF RECORD
C IFLD FIELD OF RECORD OF INTEREST
C UNIT UNIT OF MEASURE DESIRED

2 COMMON /VPROTO/
3 REAL*8 FLDNAM, RECNUM, DEFAULT
4 COMMON /NPARS/
5 COMMON /AUX/
6 REAL*8 BL8, DASHES
7 EQUIVALENCE (FNULL, NULL), (IBLS, BL8)
8
C DATA CASH /*-*/
9
10 IDVAL2 = IX
11 IFLRC = IFLCPT(ICMP+1) + IFLD
12 IF (IDVAL2 .EQ. NULL) IDVAL2 = DEFAULT(IFLRC)
13 IF (UNIT .EQ. DASH .OR. UNIT .EQ. UNITS(IFLRC) .OR. UNITS(IFLRC)
14 + .EQ. DASH) RETURN
C SEARCH TRANSLATION TABLE FOR UNIT TRANSLATION FACTOR
15 DO 10 I=1, NTRNS
16 IF (UNIT .NE. TUNITS(1, I) .OR. UNITS(IFLRC) .NE. TUNITS(2, I))
17 + GO TO A
18 IDVAL2 = IDVAL2 * TUNITS(3, I)
19 RETURN
20 A IF (UNIT .NE. TUNITS(2, I) .OR. UNITS(IFLRC) .NE. TUNITS(1, I))
UTILITY ROUTINES - IDVAL2 FUNCTION

C

+  GO TO 10
18  IDVAL2 = IDVAL2 / UNITS(3,1)
19  RETURN
20  10  CONTINUE
C
C  WRITE ERROR MESSAGE THAT UNIT TRANSLATION NOT FOUND
21  CALL EPR(37, 'IDVAL2', ICMP+1, UNIT, UNITS(IFLDRC))
22  RETURN
23  END
UTILITY ROUTINES - INITVL SUBROUTINE

1 SUBROUTINE INITVL(IVEC,N,IVAL)

C SUBROUTINE TO INITIALIZE ALL THE ELEMENTS OF AN INTEGER ARRAY TO
C A SPECIFIED VALUE.

C INPUT PARAMETERS:
C IVEC     A BLOCK OF CONTIGUOUS STORAGE TO BE INITIALIZED
C          (IT NEED NOT BE A SINGLE 1-DIM ARRAY IN THE CALLING
C          ROUTINE, THOUGH IT IS TREATED AS SUCH IN INITVL).
C N        THE NUMBER OF COMPUTER WORDS IN IVEC TO BE INITIALIZED.
C IVAL     THE VALUE TO SET EACH OF THE WORDS OF IVEC TO.

2 DIMENSION IVEC(N)
3 DO 10 I=1,N
4   IVEC(I) = IVAL
5 RETURN
6 END
UTILITY ROUTINES - INLIST FUNCTION

FUNCTION INLIST(IVAL,IFILE,NF,IFLD,JNEXT,IREC)

C ROUTINE SEARCHES A LINKED LIST OF RECORDS FOR A RECORD WITH A
C SPECIFIED VALUE IN A SPECIFIED FIELD. FUNCTION RETURNS A POINTER
C TO THE RECORD IF IT FINDS ONE, NULL OTHERWISE.

C INPUT PARAMETERS:
C IVAL VALUE TO LOOK FOR
C IFILE FILE IN WHICH RECORDS CHAINED TOGETHER ARE LOCATED
C NF NUMBER OF FIELDS IN EACH RECORD
C IFLD FIELD IN WHICH TO SEARCH FOR VALUE
C JNEXT FIELD USED FOR CHAINING RECORDS
C IREC FIRST RECORD IN LIST

COMMON /AUX/
REAL*8 BL8,DASHES
EQUIVALENCE (FNULL,NUL),(IBLS,BLS)

DIMENSION IFILE(NF,1)

INLIST = IREC
IF (INLIST .EQ. NULL) RETURN
IF (IFILE(IFLD,INLIST) .EQ. IVAL) RETURN
INLIST = IFILE(JNEXT,INLIST)
GO TO 5
END
UTILITY ROUTINES - FUNCTION IOKVAL

I SN 1 LOGICAL FUNCTION IOKVAL (IVALE, IRECTP, IFLD, ISAV)
C
C FUNCTION TO TEST WHETHER THE USER HAS SPECIFIED ANY CONSTRAINTS
C ON A PARTICULAR ATTRIBUTE'S VALUE AND IF SO WHETHER THE SPECIFIED
C VALUE SATISFIES THESE CONSTRAINTS. THE FUNCTION RETURNS .TRUE. IF NO
C CONSTRAINTS ARE SPECIFIED OR IF THE INDICATED VALUE SATISFIES ALL
C CONSTRAINTS.
C
C INPUT PARAMETERS:
C IVALUE VALUE TO BE TESTED
C IRECTP RECORD TYPE IN WHICH VALUE IS LOCATED
C IFLD FIELD OF RECORD IN WHICH VALUE LOCATED
C ISAV RECORD NUMBER WHICH RELATION REFERENCE STORED
C
COMMON /VEH/
DIMENSION IVALE(72,3), IFLD(48,9), IRECTP(42,9), FMAING(44,9),
+ ISENSR(22,9), ISTALL(32,6), IGNCTL(28,9), IAMMO(30,30),
+ IAMMOC(26,9), IENGINE(40,8), ITRANSI(32,8), IFINDR(28,10),
+ IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),
+ IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),
+ IELECT(26,5), ICCMDO(28,9), IFIREX(28,10), IENVIR(26,9),
+ IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,13), IWEYS (26,10),
+ IRELN(3,50), VFILE(1), VFILE(1)
EQUVALENCE (VEHICLE(1), IVALE(1)), (HULL(1), IFLD(1)), (TURRET(1),
+ IRECTP(1)), (MAING(1), FMAING(1)), (MACGUN1(1), FMACGN1(1)),
+ (MIGUN1(1), FMIGUN1(1)), (ADGUN1(1), IADGUN1(1)), (RANGER(1),
+ IRANGR(1)), (SENSR(1), ISENSR(1)), (STALL(1), ISTABL(1)),
+ (GIGNCTL(1), IGIGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
+ IAMMOC(1)), (ENGINE(1), IENGINE(1)), (TRANSM(1), ITRANSM(1)),
+ (FINDR(1), IFINDR(1)), (ROADW(1), IROADW(1)), (SPRING1(1),
+ ISPRNG(1)), (TRACK1(1), ITRACK1(1)), (SKIRT1(1), ISKIRT1(1)),
+ (FUEL1(1), IFUEL1(1)), (FUELC1(1), IFUELC1(1)), (CREW1(1),
+ ICREW1(1)), (CARGO1(1), ICARGO1(1)), (ELECT1(1), IELECT1(1)),
+ (COMMO1(1), ICCMDO1(1)), (FIREX1(1), IFIREX1(1)), (ENVIR1(1),
+ IENVIR1(1)),...
UTILITY ROUTINES - FUNCTION IOKVAL

**ISN**
+ [ENVIR(1)], [DIAGNS(1), IDIAGN(1)], [SIGSUP(1), SIGSP(1)],
+ [SMOKEG(1), SMOKE(1)], [EWSYS(1), IESYS(1)], [RELN(1), IRELN(1)]
5 EQUIVALENCE (VE+CLE(1), VFILE(1), IVFILE(1))
6 COMMON /VPROTN/
7 REAL*8 FLNAME, RECNAM, DEFAULT
8 COMMON /JPARN/
9 COMMON /JPARN1/
10 COMMON /AUX/
11 REAL*8 RL8, DASHES
12 EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
C 13 IOKVAL = .TRUE.
C C GET CONSTRAINT RECORD IF ONE EXISTS
14 NF = NFLDS(IRECTP)
15 IOFF = IFILPT(IRECTP)
C C TEST IF RELATION REFERENCE PASSED
C 16 IF(ISAV .EQ. NULL) GO TO 3
17 IREC=ISAV
18 GO TO 4
19 3 IREC = ISPECV
20 IF (IRECTP .GT. 1) IREC = IVECLE(IRECTP-1, ISPECV)
21 IF (IREC .EQ. NULL) RETURN
22 4 IR = IVAL(VFILE(IOFF), NF, IFLD, IREC)
23 5 IF (IR .EQ. NULL) RETURN
24 IV = IRELN(JVALUE, IR)
25 IRTYP = IRELN(JRELOP, IR)
26 GO TO (10, 15, 20, 25, 30, 35), IRTYP
27 10 IF (IVALU .LT. IV) GO TO 40
28 IOKVAL = .FALSE.
29 RETURN
30 15 IF (IVALU .LE. IV) GO TO 40
31 IOKVAL = .FALSE.
32 RETURN
UTILITY ROUTINES - FUNCTION INKVAL

ISN
33    20 IF (IVALE .EQ. IV) GO TO 40
34          IOKVAL = .FALSE.
35          RETURN
36    25 IF (IVALE .NE. IV) GO TO 40
37          IOKVAL = .FALSE.
38          RETURN
39    30 IF (IVALE .GE. IV) GO TO 40
40          IOKVAL = .FALSE.
41          RETURN
42    35 IF (IVALE .GT. IV) GO TO 40
43          IOKVAL = .FALSE.
44          RETURN

C
C GET NEXT CCNSTRAINT
45    40 IR = IRELN(JNEXT,IR)
46          GO TO 5
47          END
UTILITY ROUTINES - IRFIND FUNCTION

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>COMMON /VPRO TO/</td>
</tr>
<tr>
<td>19</td>
<td>REAL*8 FLDNAM,RECNAM,DEFAULT</td>
</tr>
<tr>
<td>20</td>
<td>COMMON /JPARS/</td>
</tr>
<tr>
<td>21</td>
<td>COMMON /JPARS1/</td>
</tr>
<tr>
<td>22</td>
<td>CMMCN /NPARS/</td>
</tr>
<tr>
<td>23</td>
<td>COMMON /OUTPLT/</td>
</tr>
<tr>
<td>24</td>
<td>LOGICAL*1 DF</td>
</tr>
<tr>
<td>25</td>
<td>REAL*8 XNAT,XMANUF,XMODEL</td>
</tr>
<tr>
<td>26</td>
<td>COMMON /AUX/</td>
</tr>
<tr>
<td>27</td>
<td>REAL*8 BL8,CASHES</td>
</tr>
<tr>
<td>28</td>
<td>EQUIVALENCE (FNULL,NULL),(IBLS,BLS)</td>
</tr>
<tr>
<td>29</td>
<td>ICMP = IRECTP - 1</td>
</tr>
<tr>
<td>30</td>
<td>IRFIND = IVECLE(ICMP,IVEH)</td>
</tr>
<tr>
<td>31</td>
<td>ARE THERE ANY COMPONENT RECORDS IN LIST ?</td>
</tr>
<tr>
<td>32</td>
<td>IF (IRFIND .EQ. NULL) RETURN</td>
</tr>
<tr>
<td>33</td>
<td>IOFF = IFILPT(IRECTP)</td>
</tr>
<tr>
<td>34</td>
<td>NF = NFLOD(IRECTP)</td>
</tr>
<tr>
<td>35</td>
<td>TEST WHETHER ID MATCHES THAT OF CURRENT RECORD</td>
</tr>
<tr>
<td>36</td>
<td>KID = IVAL(VFILE(IOFF),NF,JID,IRFIND)</td>
</tr>
<tr>
<td>37</td>
<td>IF (ID .EQ. KID) RETURN</td>
</tr>
<tr>
<td>38</td>
<td>TRY NEXT RECCPD</td>
</tr>
<tr>
<td>39</td>
<td>IRFIND = IVAL(VFILE(IOFF),NF,JNEXT,IRFIND)</td>
</tr>
<tr>
<td>40</td>
<td>IF (IRFIND .NE. NULL) GO TO 10</td>
</tr>
<tr>
<td>41</td>
<td>RETURN</td>
</tr>
<tr>
<td>42</td>
<td>END</td>
</tr>
</tbody>
</table>
UTILITY ROUTINES - ITABLE FUNCTION

ISN
1
FUNCTION ITABLE(LIST,KEY,NLIST,NKEY)

C
C ROUTINE TO SEARCH A TABLE FOR A SPECIFIED KEY. IF THE KEY IS
C PRESENT THE ROUTINE RETURNS ITS INDEX IN THE TABLE. IF THE ROUTINE
C ENCOUNTERS A BLANK SLOT OF THE TABLE IT STORES THE KEY THERE AND
C RETURNS ITS INDEX. IF IT REACHES THE END OF THE TABLE BEFORE EITHER
C FINDING THE KEY OR A BLANK SLOT, IT RETURNS A NULL VALUE.
C
C INPUT PARAMETERS:
C LIST LIST OF ITEMS TO SEARCH
C KEY KEY TO LOCK FOR
C NLIST SIZE OF LIST
C NKEY NUMBER OF WORDS IN EACH ITEM IN LIST & IN KEY
C
2
DIMENSION KEY(NKEY),LIST(NKEY,NLIST)
3
DATA NULL/-1/,IBLS/-1/
4
DO 20 I=1,NLIST
5
   DO 10 J=1,NKEY
6
      IF (KEY(J) .EQ. LIST(J,I)) GO TO 10
7
   CONTINUE
8
   ITABLE = I
9
   RETURN
10
10
DO 15 J=1,NKEY
11
   IF (LIST(J,I) .EQ. IBLS) GO TO 20
12
15
   CONTINUE
13
DO 18 J=1,NKEY
14
18
   LIST(J,I) = KEY(J)
15
   ITABLE = I
16
   RETURN
17
20
   CONTINUE
18
   ITABLE = NULL
19
   RETURN
UTILITY ROUTINES - IVAL FUNCTION

FUNCTION IVAL(IFILE,NFLD,IFLD,IREC)

FUNCTION TO RETRIEVE AND RETURN THE VALUE OF A SPECIFIED INTEGER FIELD OF A SPECIFIED RECORD IN A SPECIFIED FILE (ARRAY).

INPUT PARAMETERS:

IFILE  FILE IN WHICH VALUE IS LOCATED
NFLD   NUMBER OF FIELDS IN EACH RECORD OF FILE
IFLD   FIELD OF INTEREST
IREC   RECORD OF INTEREST

DIMENSION IFILE(NFLD,1)

IVAL = IFILE(IFLD,IREC)

RETURN
END
UTILITY ROUTINES - IZERO SUBROUTINE

1 SUBROUTINE IZERO(IARRAY,N)
   C
   C INITIALIZES AN INTEGER ARRAY TO ZERO
   C
2 DIMENSION IARRAY(N)
3   DO 10 I=1,N
4   10   IARRAY(I) = 0
5 RETURN
6 END
UTILITY ROUTINES - LSEARCH FUNCTION

ISN
1
FUNCTION LSEARCH(LIST, KEY, NLIST, NKEY)
C
FUNCTION TO SEARCH A LIST (VECTOR) FOR A SPECIFIED KEY. THE
FUNCTION RETURNS THE INDEX OF THE KEY IN THE LIST IF SUCCESSFUL, OR
A NULL VALUE IF NOT.
C
INPUT PARAMETERS:
C LIST LIST OF ITEMS TO SEARCH
C KEY KEY TO LOOK FOR
C NLIST SIZE OF LIST
C NKEY NUMBER OF WORDS CONTAINED BY EACH ITEM OF LIST & BY KEY
C
2 DIMENSION KEY(NKEY), LIST(NKEY, NLIST)
3 DATA NULL/-1/
C
4 DO 10 I=1, NLIST
5 DO 5 J=1, NKEY
6 IF (KEY(J) NE. LIST(J, I)) GO TO 10
7 5 CONTINUE
8 LSEARCH = I
9 RETURN
10 10 CONTINUE
11 LSEARCH = NULL
12 RETURN
13 END
UTILITY ROUTINES - NEWREC FUNCTION

I SN
1 FUNCTION NEWREC(IRECTP)
2 C
3 C ROUTINE FOR OBTAINING THE NEXT AVAILABLE RECORD OF THE
4 C SPECIFIED TYPE FROM THE AVAILABLE SPACE LIST AND RETURNING
5 C A POINTER TO IT.
6 C
7 C INPUT PARAMETERS:
8 C IRECTP RECORD TYPE OF INTEREST
9 C COMMON /VPROTO/
10 REAL*8 FLONAM,RECNAM,DEFAULT
11 COMMON /NPARS/

12 IF (IRECTP .LE. 1 .OR. IRECTP .GT. NRECTP) CALL ERR(7,
13 + 'NEWREC',IRECTP,0,0)
14 IF (LSTREC(IRECTP) .GE. NRECS(IRECTP)) CALL ERR(8,'NEWREC',
15 + IRECTP,NRECS(IRECTP),0)
16 LSTREC(IRECTP) = LSTREC(IRECTP) + 1
17 NEWREC = LSTREC(IRECTP)
18 RETURN
19 END
UTILITY ROUTINES - NEWREL FUNCTION

FUNCTION NEWREL(I RECTP, I FLD, K REL, I VALUE, ID)

FUNCTION TO GET A NEW RELATION RECORD AND STORE IT AS
THE VALUE (OR ONE OF A LIST OF VALUES) OF A SPECIFIED FIELD
IN A SPECIFIED COMPONENT RECORD OF THE USER SPECIFICATION VEHICLE.

INPUT PARAMETERS
I RECTP RECORD TYPE
IFLD FIELD INDEX OF THE COMPONENT TYPE
K REL AN INTEGER FROM 1 TO 6 INDICATING THE TYPE OF RELATION
I VALUE THE VALUE FIELD OF THE RELATION
ID COMPONENT IDENTIFIER IF RELATION REFERS TO A COMPONENT

COMMON /VEH/
DIMENSION IVECLE(72, 3), IHULL(48, 9), I TURET(42, 9), FMAING(44, 9),
+ FMACGN(40, 10), FMISGN(40, 7), IADGN(40, 6), IRANGR(28, 7),
+ ISENSR(32, 9), I STABL(30, 6), IGNCTL(28, 9), IAMMO(3), 30),
+ IAMMOC(26, 9), IENGINE(40, 8), ITRANS(32, 8), IFINDR(28, 10),
+ IROADW(36, 10), ISPRNG(30, 10), ITRACK(34, 8), ISKIRT(28, 8),
+ IFUEL(28, 11), IFUEL(26, 9), ICREW(26, 5), ICARGO(26, 5),
+ IELECT(26, 6), ICOMM(28, 9), IFIREX(28, 10), IENVIR(26, 9),
+ IDIAG(26, 5), ISIGSP(26, 24), I SMOKE(26, 10), I EWSYS(26, 10),
+ IRELN(3, 50), I FILE(1), I FILE(1)

EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURET(1),
+ I TURET(1)), (MAINGN(1), FMAING(1)), (AMACGN(1), FMACGN(1)),
+ (FMACGN(1), FMISGN(1), FAEGUN(1), IADGN(1)), (RANGER(1),
+ IRANGR(1)), (SENSR(1), ISENSR(1)), (STABL(1), I STABL(1)),
+ (GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
+ IAMMOC(1)), (ENGINE(1), IENGINE(1)), (TRANSM(1), ITRANS(1)),
+ (FINDR(1), IFINDR(1)), (ROADWH(1), IROADW(1)), (SPRING(1),
+ ISPRNG(1)), (TRACK(1), ITRACK(1)), (ISKIRT(1), ISKIRT(1)),
+ (FUEL(1), IFUEL(1)), (FUEL(1), IFUEL(1)), (CREW(1),
+ ICREW(1)), (ICARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
+ (COMM(1), ICOMM(1)), (FIREX(1), IFIREX(1)), (ENVIR(1),
+ IENVIR(1)), (DIAGN(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
+ IRELN(3, 50), I FILE(1), I FILE(1)
UTILITY ROUTINES - NEWREL FUNCTION

ISN
+ (SMCKEG(1),ISMCKE(1)),(EWSYS(1),IΕWSYS(1)),(RELN(1),IRELN(1))
EQUVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))
COMMNC /VPROTO/
REAL*8 FLDNAM,RECNAM,DEFAULT
COMMON /JPARS/
COMMON /JPARS1/
COMMON /NPARS/
COMMON /AUX/
REAL*8 BLB,CASHES
EQUVALENCE (FNULL,NULL),(IBLS,BLS)

C GET A NEW RELATION RECORD, IF THERE IS ONE.
IF (LSTREC(NRECTP) .GE. NRECS(NRECTP)) CALL ERR(30,'NEWREL',
+ LSTREC(NRECTP),0,0)
LSTREC(NRECTP) = LSTREC(NRECTP) + 1
NEWREL = LSTREC(NRECTP)
IOFF = IFILPT(IRECTP)
NF = NFLOS(IRECTP)
ICMP = IRECTP - 1
IREC = ISPECV
IF (IRECTP .EQ. 1) GO TO 1
IREC = IVECLE(ICMP,ISPECV)
2 IF (IREC .NE. NULL) GO TO 5
C NO COMPONENT RECORD OF THIS TYPE MATCHING THE APPROPRIATE ID
C CURRENTLY EXISTS FOR THE USER
C SPECIFICATION VEHICLE. CREATE ONE TO ATTACH RELATION RECORD TO.
IREC = NEWREC(IRECTP)
CALL STORE(IVECLE(ICMP,ISPECV),VFILE(IOFF),NF,JNEXT,IREC)
IVECLE(ICMP,ISPECV) = IREC
C STORE COMPONENT RECORD ID
CALL STORE(10,VFILE(IOFF),NF,JID,IREC)
GO TO 15
C IS THIS RECORD APPROPRIATE FOR ATTACHING CONSTRAINT TO ?
5 KID = IVAL(VFILE(IOFF),NF,JID,IREC)
IF (KID .NE. 10) GO TO 8
UTILITY ROUTINES - NEWREL FUNCTION

C IF SO, CREATE A NEW RELATION RECORD AND STORE THE POINTER IN THE
C APPROPRIATE FIELD. IF A PREDEFINED ALTERNATE COMPONENT IS TO BE
C CHANGED, WARN THE USER.
31 IF (ID .GT. NALTCS(ICMP)) GO TO 15
32 CALL WARN(1,'NEWREL',IRECTP,IFLD,ID)
33 CALL STORE(IValue,VFILE(IOFF),NF,IFLD,IREC)
34 RETURN
C IF NOT, TRY OTHER COMPONENT RECORDS
35 IREC = IVAL(VFILE(IOFF),NF,JNEXT,IREC)
36 GO TO 2
37 10 IRELN(JNEXT,NEWREL) = IVECLE(IFLD,ISPECV)
38 15 IF (IRECTP .NE. 1) IRELN(JNEXT,NEWREL) =
  + VFILE(IOFF + (NF * (IREC-1)) + (IFLD-1))
39 CALL STORE(NEWREL,VFILE(IOFF),NF,IFLD,IREC)
40 IRELN(JRELP,NEWREL) = KREL
41 IRELN(IValue,NEWREL) = IValue
42 RETURN
43 END
UTILITY ROUTINES - NEWVEH FUNCTION

1  FUNCTION NEWVEH(I1,I2,I3,I4)

C FUNCTION TO GET A NEW RECORD IN IRECS AND STORE
C INFORMATION IN IT. THE FUNCTION RETURNS A POINTER TO THE
C RECORD OBTAINED.
C
C INPUT PARAMETERS:
C I1 VALUE OF 1ST FIELD
C I2 VALUE OF 2ND FIELD
C I3 VALUE OF 3RD FIELD
C I4 VALUE OF 4TH FIELD
C
2  COMMON /SPECS2/
3  REAL*8 VEHS

4  LSTVEH = LSTVEH + 1
5  IF (LSTVEH .GT. NVR) CALL ERR(21,'NEWVEH',NVR,I4,0)
6  NEWVEH = LSTVEH
7  IRECS(JNXT,LSTVEH) = I1
8  IRECS(JTYP,LSTVEH) = I2
9  IRECS(JSPEC,LSTVEH) = I3
10  IRECS(JVEH,LSTVEH) = I4
11  RETURN
12  END
UTILITY ROUTINES - NLIST FUNCTION

FUNCTION NLIST(IFILE,NF,JNEXT,ISAVE)

C ROUTINE TO COUNT THE NUMBER OF RECORDS CHAINED TOGETHER IN A LIST.

C INPUT PARAMETERS:
C IFILE  FILE CONTAINING THE RECORDS
C NF    NUMBER OF FIELDS OF FILE
C JNEXT NEXT POINTER
C ISAVE 1ST RECORD OF LIST

COMMON /AUX/
PEAL*8 BL8,CASHES
EQUIVALENCE (FNULL,NULL),(IBLS,BLS)

DIMENSION IFILE(NF,1)
IR=ISAVE

NLIST = 0
IF (IP .EQ. NULL) RETURN
IR = IFILE(JNEXT,IP)
NLIST = NLIST + 1
GO TO 5
END
UTILITY ROUTINES - NULIFY SUBROUTINE

1 SUBROUTINE NULIFY(IBLOCK,N)

C ROUTINE TO NULL OUT THE CONTENTS OF A BLOCK OF STORAGE.

C INPUT PARAMETERS:
C IBLOCK LOCATION OF THE BEGINNING OF THE BLOCK TO BE NULLED.
C N SIZE OF THE BLOCK TO BE NULLED

2 DIMENSION IBLOCK(N)
3 DATA NULL/-1/

4 DO 10 I=1,N
5 10 IBLOCK(I) = NULL
6 RETURN
7 END
UTILITY ROUTINES - FUNCTION OKVAL

ISN

LOGICAL FUNCTION OKVAL(VAL, IRECP, IFLD, ISAV)

C
C FUNCTION TO TEST WHETHER THE USER HAS SPECIFIED ANY CONSTRAINTS
C ON A PARTICULAR ATTRIBUTE'S VALUE AND IF SO WHETHER THE SPECIFIED
C VALUE SATISFIES THESE CONSTRAINTS. THE FUNCTION RETURNS TRUE IF NO
C CONSTRAINTS ARE SPECIFIED OR IF THE INDICATED VALUE SATISFIES ALL
C CONSTRAINTS.

C INPUT PARAMETERS:
C VAL VALUE TO BE TESTED
C IRECP RECORD TYPE IN WHICH VALUE IS LOCATED
C IFLD FIELD OF RECORD IN WHICH VALUE LOCATED
C ISAV RECORD NUMBER IF RELATION RECORD PASSED

COMMON /VEH/
DIMENSION IVECLE(72, 3), IHULL(48, 9), ITURET(42, 9), FMAING(44, 9),
+ FMACGN(40, 10), FMISGN(40, 7), IADGN(40, 6), IRANGR(28, 7),
+ ISENS(32, 9), ISTABL(30, 6), IGNCNT(28, 9), IAMMO(30, 30),
+ IAMMOC(26, 9), IENGIN(40, 8), ITRANS(32, 8), IFINDR(28, 10),
+ IRODW(36, 10), ISPRNG(33, 10), ITRACK(34, 8), ISKIRT(28, 8),
+ IFUEL(28, 10), IFUELC(26, 9), ICREW(26, 5), ICARGO(26, 5),
+ IELECT(26, 6), ICOMM(26, 9), IFIREX(28, 10), IENVIR(26, 9),
+ IDIGN(26, 5), ISIGSP(26, 24), ISMOKE(26, 10), IWEYS(26, 10),
+ IRELN(3, 50), VFILE(1), IFFILE(1)

EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
+ ITURET(1)), (MAINGN(1), FMAING(1)), (MAGCN(1), FMACGN(1)),
+ (MISGN(1), FMISGN(1)), (ADGN(1), IADGN(1)), (RANGER(1),
+ IRANGR(1)), (SENS(1), ISENS(1)), (STABLE(1), ISTATBLE(1)),
+ (UNCT(1), IGNCNT(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
+ IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANS(1), ITRANS(1)),
+ (FINLDR(1), IFINDR(1)), (ROADW(1), IRODW(1)), (SPRING(1),
+ ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),
+ (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
+ ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECTR(1)),
+ (COMM(1), ICOMM(1)), (FIREX(1), IFIREX(1)), (ENVIR(1),
+ IENVIR(1))
UTILITY ROUTINES - FUNCTION OKVAL

ISN

+ TENVIR(1), (DIAGNS(1), IDIAGN(1), (SIGSUP(1), IS(GSP(1)),
+ (SMOKEG(1), ISMOKE(1)), (EWSYS(1), IEWYS(1)), (RELN(1), IRELN(1))
5
EQUIVALENCE (VEICLE(1), VFILE(1), IFILE(1))
6
COMMON /VPROTO/
7 REAL*8 FLDNAM, RECNAM, DEFAULT
8 COMMON /JPARS/
9 CCMDON /JPARSIL/
10 COMMON /AUX/
11 REAL*8 BL8,CASES
12 EQUIVALENCE (FNULI, NULL), (IBLS, BL8)
C
13 CKVAL = .TRUE.
C
GET CONSTRAINT RECORD IF ONE EXISTS
14 NF = NFLDS (IRECTP)
15 IOFF = IFILPT (IRECTP)
C
TEST IF RELATION REFERENCE PASSED
C
16 IF (ISAV .EQ. NULL) GO TO 3
17 IPFC = ISAV
18 GO TO 4
19 3 IRFC = ISPFCV
20 IF (IRECTP .LT. 1) IREC = IVECLE (IRECTP-1, ISPECV)
21 IF (IREC .EQ. NULL) RETURN
22 4 IR = IVAL (VFILE (IOFF), NF, IFLD, IREC)
23 5 IF (IR .EQ. NULL) RETURN
24 V = RELN (JVALUE, IR)
25 IRTYP = IRELN (JRELON, IR)
26 GO TO (10, 15, 20, 25, 30, 35), IRTYP
27 10 IF (VAL .LT. V) GO TO 40
28 CKVAL = .FALSE.
29 RETURN
30 15 IF (VAL .LE. V) GO TO 40
31 OKVAL = .FALSE.
32 RETURN
UTILITY ROUTINES - FUNCTION OKVAL

ISN
33  20  IF (VAL .EQ. V) GO TO 40
34       OKVAL = .FALSE.
35       RETURN
36  25  IF (VAL .NE. V) GO TO 40
37       OKVAL = .FALSE.
38       RETURN
39  30  IF (VAL .GE. V) GO TO 40
40       OKVAL = .FALSE.
41       RETURN
42  35  IF (VAL .GT. V) GO TO 40
43       OKVAL = .FALSE.
44       RETURN

C
C GET NEXT CONSTRAINT
45  40  IR = IRELN(JNEXT,IR)
46       GO TO 5
47       END
UTILITY ROUTINES - QUEUE SUBROUTINE

SUBROUTINE QUEUE(IR, LISTPT, IRECTP)

ROUTINE FOR QUEUING A RECORD ONTO A LIST.

INPUT PARAMETERS:
IR     THE RECORD TO BE QUEUED
LISTPT A POINTER TO THE HEAD OF THE LIST
IRECTP THE TYPE OF RECORD

COMMON /VFEH/
DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAINING(44,9),
+ FMCGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
+ ISENSR(32,9), ISTABL(30,6), IGNCTR(28,9), IAMMO(30,30),
+ IAMMOC(26,9), IENGINE(40,8), ITRANS(32,8), IFINDR(28,10),
+ IROADW(36,10), ISPRNG(30,10), ITASK(34,8), ISKIRT(28,9),
+ IFUEL(28,10), IFUEL(26,9), ICREW(26,5), ICARGO(26,5),
+ IELECT(26,6), ICOMM(28,9), IFIREX(28,10), IENVIR(26,9),
+ IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
+ IRELN(3,50), VFILE(1), IVFILE(1)

EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
+ ITURET(1)), (MAINGN(1), FMAINING(1)), (MACGUN(1), FMCGN(1)),
+ (MISGN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGR(1),
+ IRANGR(1)), (SENSR(1), ISENSR(1)), (STABLE(1), ISTABL(1)),
+ (GUNCTL(1), IGNCTR(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
+ IAMMOC(1)), (ENGINE(1), IENGINE(1)), (TRANS(1), ITRANS(1)),
+ (FINLDR(1), IFINDR(1)), (ROADW(1), IROADW(1)), (SPRNG(1),
+ ISPRNG(1)), (TRK(1), ITRK(1)), (SKIRT(1), ISKIRT(1)),
+ (FUEL(1), IFUEL(1)), (IFUEL(1), IFUEL(1)), (CREW(1),
+ ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
+ (COMMO(1), ICOMM(1)), (FIREX(1), IFIREX(1)), (ENVIR(1),
+ IENVIR(1)), (DIAGN(1), IDIAGN(1)), (SIGSP(1), ISIGSP(1)),
+ (SMCKEG(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))

EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))

COMMON /VPROTO/
REAL*8 FLNAM, RECNUM, DEFAULT
UTILITY ROUTINES - QUEUE SUBROUTINE

10 COMMON /JPARS/
11 REAL*8 BLA,BLASH
12 EQUIVALENCE (FNULL,NULL),(IBLS,BLS)

C
13 IOFF = TFILP(IRECTP)
14 NF = NFLTDS(IRECTP)
15 IF (LISTPT .NE. NULL) GO TO 10
16 LISTPT = IR
17 CALL STORE(NULL,VFILE(IOFF),NF,JNEXT,IR)
18 C RETURN

C
19 10 IREC = LISTPT
C GET VALUE OF NEXT FIELD OF RECORD
20 IRECL = IREC
21 IREC = TVAL(VFILE(IOFF),NF,JNEXT,IREC)
22 IF (IREC .NE. NULL) GO TO 15
23 CALL STORE(IR,VFILE(IOFF),NF,JNEXT,IRECL)
24 CALL STORE(NULL,VFILE(IOFF),NF,JNEXT,IR)
25 RETURN
26 END
UTILITY ROUTINES - STONAM SUBROUTINE

1 SUBROUTINE STONAM(CNAME, IVFLAG)

C ROUTINE TO STORE AN OUTPUT NAME READ FROM THE PROTOTYPE
C DEFINITION FILE IN THE NEXT AVAILABLE RECORD OF THE DJTNAM ARRAY.

C INPUT PARAMETERS:
C CNAME NAME TO BE STORED
C IVFLAG AN INDICATOR WHICH HAS THE VALUE (0) IF THERE IS NO VALUE
C TO PRINT. IF THERE IS A VALUE TO PRINT THEN IFLDRC IS THE
C INDEX OF THE FIELD IN THE RECORD.

2 COMMON /VPROTO/
3 REAL*8 FLDNAM, RCNAM, DFUALT
4 COMMON /NPARS/
5
6 DIMENSION CNAME(NOUTWC)

7 LSTOUT = LSTOUT + 1
8 IF (LSTOUT .GT. NOUT) CALL ERR(31, 'STONAM', NOUT, 0, 0)
9 CALL COPY(CNAME, OUTNAM(1, LSTOUT), NOUTWD)
10 IOUTF(LSTOUT) = IVFLAG
11 RETURN
12 END
UTILITY ROUTINES – STORE SUBROUTINE

1
SUBROUTINE STORE(VALUE, FILE, NFLD, IFLD, IREC)
C
C ROUTINE TO STORE A VALUE IN A PARTICULAR FIELD OF A
C PARTICULAR RECORD OF A SPECIFIED FILE (ARRAY).
C
C INPUT PARAMETERS:
C VALUE VALUE TO BE STORED
C FILE FILE IN WHICH TO STORE VALUE
C NFLD NUMBER OF FIELDS IN EACH RECORD OF FILE
C IFLD FIELD IN WHICH TO STORE VALUE
C IREC RECORD IN WHICH TO STORE VALUE
C
2 DIMENSION FILE(NFLD,1)
C
3 FILE(IFLD,IREC) = VALUE
4 RETURN
5 END
UTILITY ROUTINES - SUM FUNCTION

FUNCTION SUM(X,N)

C THIS SUM.

DIMENSION X(N)

SUM = 0.

DO 10 I = 1,N

10 SUM = SUM + X(I)

RETURN END

ISN
UTILITY ROUTINES - SUMCMP FUNCTION

1 FUNCTION SUMCMP(IV,IFLO,UNIT)

2 COMMON /VEH/

3 DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
   + FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
   + ISENS(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
   + IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
   + IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
   + IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
   + IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
   + IDIAGN(26,51),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
   + JRELN(3,50),VFILE(1),IVFILE(1)

4 EQUIVALENCE (VECLE(1),IVECLE(1)),(HUILL(1),IHULL(1)),(TURET(1),
   + ITURET(1)),(MAINGN(1),FMAING(1)),(MACGN(1),FMACGN(1)),
   + (MISGN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
   + TRANGR(1)),(SENSR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
   + (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
   + IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANS(1),ITRANS(1),
   + (FINDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
   + SPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
   + (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
   + CREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1),
   + (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIR(1),
   + ENVIR(1)),(DIAGN(1),IDIAGN(1),(SIGSUP(1),ISIGSP(1)),
   + (SMOKE(1),ISMOKE(1)),(EWY(1),IEWYS(1)),(RELN(1),JRELN(1))

5 EQUIVALENCE (VECLE(1),VFILE(1),IVFILE(1))

6 COMMON /VPPORO/
UTILITY ROUTINES - SUMCMP FUNCTION

7 REAL*8 FLDNAM,RECNAME,DEFAULT
8 COMMON /JPARS/
9 COMMON /JPARSI/
10 COMMON /NPARS/
11 COMMON /AUX/
12 REAL*8 BLR,DASHES
13 EQUIVALENCE (FNULL,NULL),(IBLS,ALS)

C
14 SUMCMP = 0.
15 DO 20 ICMP=1,NCMPTOP
16 IRECTP = ICMP + 1
17 NF = NFLOD(IRECTP)
18 IOFF = IFILPT(IRECTP)
19 IREC = IVECLE(ICMP,IV)
20 10 IF (IREC .EQ. NULL) GO TO 20
21 SUMCMP = SUMCMP + DVAL(VFILE(IOFF),NF,IFLD,IREC,IRECTP,UNIT,DF)
22 IREC = IVAL(VFILE(IOFF),NF,JNEXT,IREC)
23 GO TO 10
24 20 CONTINUE
25 RETURN
26 END
UTILITY ROUTINES - SUMPD2 FUNCTION

FUNCTION SUMPD2(IX,Y,N)

C FUNCTION TO CALCULATE THE SUM OF THE PRODUCT OF CORRESPONDING
C ELEMENTS IN TWO EQUAL SIZED ARRAYS.
C
C COMPUTES THE VECTOR PRODUCT IX*Y.
C
DIMENSION IX(N),Y(N)
SUMPD2 = 0.
DO 10 I=1,N
10 SUMPD2 = SUMPD2 + IX(I)*Y(I)
RETURN
END
UTILITY ROUTINES - SUMPRD FUNCTION

```
1 FUNCTION SUMPRD(X,Y,N)
   C
   C FUNCTION TO CALCULATE THE SUM OF THE PRODUCT OF CORRESPONDING
   C ELEMENTS IN TWO EQUAL SIZED ARRAYS.
   C
   C COMPUTES THE VECTOR PRODUCT X*Y.
   C
   DIMENSION X(N),Y(N)
   SUMPRD = 0.
   DO 10 I=1,N
   10 SUMPRD = SUMPRD + X(I)*Y(I)
   RETURN
   END
```
UTILITY ROUTINES - TRANSF FUNCTION

FUNCTION TRANSF(VALUE, MULOP, FACTOR, ADDOP, TERM)

C ROUTINE TO TRANSFORM A VALUE BY MULTIPLYING IT BY A CONSTANT
C AND/OR BY ADDING OR SUBTRACTING A CONSTANT FROM IT, IF SUCH
C OPERATIONS ARE SPECIFIED BY THE USER. THE FUNCTION RETURNS THE
C TRANSFORMED VALUE.
C
C INPUT PARAMETERS:
C VALUE VALUE TO BE TRANSFORMED (POSSIBLY)
C MULOP MULTIPLY OPERATOR (OR BLANK)
C FACTOR CONSTANT TO BE MULTIPLIED
C ADDOP + OR - OPERATOR (OR BLANK)
C TERM CONSTANT TO BE ADDED OR SUBTRACTED
C
COMMON /AUX/
REAL*8 BLS, DASHES
EQUIVALENCE (FNULL, NULL), (IBLS, BLS)

TRANSF = VALUE
IF (MULOP .NE. MULT) GO TO 10
TRANSF = TRANSF * FACTOR

10 IF (ADDOP .NE. PLUS) GO TO 20
TRANSF = TRANSF + TERM
RETURN

20 IF (ADDOP .NE. SMINUS) RETURN
TRANSF = TRANSF - TERM
RETURN

END
UTILITY ROUTINES - UNHOOK SUBROUTINE

SUBROUTINE UNHOOK(ICMP)

ROUTINE TO UNCHAIN COMPONENT RECORDS OF A GIVEN TYPE CHAINED TOGETHER AS PART OF THE SPECIFICATION VEHICLE DESCRIPTION WHENEVER BACKTRACKING IS NECESSARY.

INPUT PARAMETERS:

ICMP   COMPONENT TYPE

COMMON /VEH/

DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
+ FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
+ ISNSR(32,9), ISTATL(3,6), IIGNCT(28,9), IAMMO(3,30),
+ IAMMAC(26,9), IENGIN(40,8), ITTRANS(32,8), IFINDR(28,10),
+ IROADW(36,10), ISPRNG(30,10), ITPACK(34,8), ISKIRT(28,8),
+ IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),
+ ISELECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),
+ IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IIEWSYS(26,10),
+ IRELN(3,5), IVFIE(1), IVFILE(1)

EQUIVALENCE (VECLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
+ ITURET(1)), (MAINGN(1), FMAING(1)), (MACGN(1), FMACGN(1)),
+ (MISGN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
+ IRANGR(1)), (SENSR(1), ISENSR(1)), (STABLE(1), ISTATL(1)),
+ (IGNCT(1), IIGNCT(1)), (AMMO(1), IAMMO(1)), (AMMAC(1),
+ IAMMAC(1)), (ENGINE(1), IENGIN(1)), (TRANS(1), ITRANS(1)),
+ (FINDN(1), IFINDR(1)), (ROADW(1), IROADW(1)), (SPRING(1),
+ ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),
+ (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
+ ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), ISELECT(1)),
+ (COMMO(1), ICOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIR(1),
+ IENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSP(1), ISIGSP(1)),
+ (SMOKE(1), ISMOKE(1)), (EWSYS(1), IIEWSYS(1)), (RELN(1), IRELN(1))

EQUIVALENCE (VECLE(1), IVFIE(1), IVFILE(1))

COMMON /VPRCTC/

REAL*8 FLDMHAM, RECNAM, DEFAULT
UTILITY ROUTINES - UNHOOK SUBROUTINE

COMMON /JPARS/
COMMON /JPARS1/
COMMON /AUX/
REAL*8 BL8, CASHES
EQUIVALENCE (FNUL, NULL), (IBLS, BLS)

C
IR = IVECLE(ICMP, ISPECV)
IF (IR .EQ. NULL) RETURN
IVECLE(ICMP, ISPECV) = NULL
IRECTP = ICMP + 1
NF = NFLDS(IRECTP)
IOFF = IFILPT(IRECTP)
NEXT = IVAL(VFILE(IOFF), NF, JNEXT, IR)
CALL STORE(NULL, VFILE(IOFF), NF, JNEXT, IR)
IF (NEXT .EQ. NULL) RETURN
IR = NEXT
GO TO 10
END
UTILITY ROUTINES - ZERO SUBROUTINE

1 SUBROUTINE ZERO(V,N)
   C INITIALIZES AN ARRAY TO ZERO
   C
2 DIMENSION V(N)
3 DO 10 I=1,N
4   V(I) = 0.
5 RETURN
6 END
# DISTRIBUTION LIST

Please notify USATARADCOM, DROTA-ZC, Warren, MI 48090, of corrections and/or changes in address.

<table>
<thead>
<tr>
<th>No. of Copies</th>
<th>Name and Title</th>
<th>Organization and Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Commander</td>
<td>US Army Tank-Automotive Research and Development Command Attention: DRDTA-UL Warren, MI 48090</td>
</tr>
<tr>
<td>17</td>
<td>Commander</td>
<td>US Army Tank-Automotive Research and Development Command Attention: DRDTA-Z Warren, MI 48090</td>
</tr>
<tr>
<td>1</td>
<td>Director</td>
<td>US Army Human Engineering Laboratory Attention: Mr. A. Eckels Aberdeen Proving Grounds, MD 21005</td>
</tr>
<tr>
<td>2</td>
<td>Director</td>
<td>US Army Ballistic Research Laboratory Attention: Dr. A. Rainis, Mr. G. Bowers Aberdeen Proving Grounds, MD 21005</td>
</tr>
<tr>
<td>2</td>
<td>Director</td>
<td>Defense Documentation Center Cameron Station Alexandria, VA 22314</td>
</tr>
</tbody>
</table>
This report describes a parametric engineering system definition model developed for use in planning tank-automotive research and development. The model can be used for (1) estimating the performance of a conceptual armored combat vehicle consisting of a specified set of components or (2) "sizing" a vehicle to meet a set of performance specifications. Key elements of the model include a structure for a data base to contain descriptions of components which might be incorporated into future armored combat vehicles.
combat vehicles, (2) a solution algorithm which uses a combinatorial approach to search over alternative combinations of components to find one which meets specifications input by the model user, (3) a variety of engineering relationships and "look-up table" functions for estimating system engineering parameters and performance characteristics, and (4) routines which output a description of the concept vehicle generated by the model.