Appendix I

SAMPLE ROUTINE LISTINGS

I.1 Listing of Subroutine skr_turn_on.f

Notice that format 6005 is not used, there is no information under the VARIABLE DESCRIPTIONS heading, and there are some write (6, ---) statements. The comments here are lower case which makes them easy to read.

```
      subroutine skr_turn_on(iskr)
C*** ----------------------------------------------------------------
C*** THIS SUBROUTINE IS PART OF THE V6.2 VERSION OF BRAWLER.
C*** THIS SUBROUTINE CONTAINS MATERIAL FURNISHED TO THE U.S. GOVERNMENT
C*** WITH UNLIMITED RIGHTS UNDER CONTRACT F49650-89-D5033, BY:
C***    DECISION-SCIENCE APPLICATIONS, INC.
C***    1110 NORTH GLEBE ROAD, SUITE #400,
C***    ARLINGTON, VA 22201.  (703)-243-2500
C*** ----------------------------------------------------------------
C#ABSTRACT Changes a missile seeker from 'off' to 'on'
C#PURPOSE  Called during missile fly event to see if seeker has
C          met its criteria for turning on and attempting to make
C          observations
C#AUTHOR   Eiserman   16-Jan-1991
C#TYPE     Missile
C#PARAMETER DESCRIPTIONS
CIN  ISKR               INT - Seeker index in /misdat/, /mslflt/
C#TECHNICAL DESCRIPTION
C Different seekers have different criteria for turning on when a
C missile is launched with the seeker off (i.e. in command guided mode
C or a multi-seeker missile with some other seeker(s) on at launch).
C This routine checks the seeker sequence algorithm to see if it is
C time to attempt to turn the seeker on, then checks the seeker type
C and applies the appropriate test.
C Note that nothing is done if the seeker is already on.
C This routine assumes that the data for the indicated seeker is
C already current.
C#VARIABLE DESCRIPTIONS
C !                  !
C####
C#AUDIT
C **VERSION V6.2** 12-Oct-95----03:22 DSA -> SURVIAC SITE1 002 SILICONG
C PART-TASK BY ELazarus ON 03-Aug-95 11:00:32 Thu FROM TASK new_guid
C MODIFIED BY ELazarus ON 21-Jul-95  11:51:09 Fri FOR TASK new_guid
C  Handle no_data=.true. return from msl_gui_pos
C PART-TASK BY ELazarus ON 17-Jul-95 19:28:45 Mon FROM TASK new_guid
C MODIFIED BY ELazarus ON 02-Jun-95  12:13:14 Fri FOR TASK new_guid
C  Now checking hav_rng for each seeker type
C PART-TASK BY ELazarus ON 24-Aug-94 17:01:10 Wed FROM TASK for62
C MODIFIED BY ELazarus ON 05-Aug-94  11:20:23 Fri FOR TASK for62
C  Call add_act2bb instead of having code in-line.
C PART-TASK BY ELazarus ON 20-May-94 15:14:46 Fri FROM TASK for62
```

```
C MODIFIED BY ELazarus ON 20-May-94  12:06:11 Fri FOR TASK for62
C  Added lprnt3 with AFSAA prints.
C PART-TASK BY GKEiserman ON 24-May-93 09:53:21 Mon FROM TASK bugfix7
C   Associated with bugfix form 93-0918
C MODIFIED BY GKEiserman ON 25-May-93  08:36:40 Mon FOR TASK bugfix7
C  Fixed code to make sure skr_name, skr_len are set before they are used.
C PART-TASK BY Kramer ON 11-Mar-93 12:18:42 Thu FROM TASK bugfix8
C   Associated with bugfix form 93-0869
C MODIFIED BY Kramer ON 10-Mar-93  11:07:50 Wed FOR TASK bugfix8
C  Extended lprnts.
C PART-TASK BY Eiserman ON 03-Aug-92 16:40:49 Mon FROM TASK bugfix10
C   Associated with bugfix form 92-0684
C MODIFIED BY Eiserman ON 03-Aug-92  16:19:16 Mon FOR TASK bugfix10
C  Moved declaration of iside to unclassified section.
C####
#include "par"
#include "pcon"
#include "msl_par"
#include "arodat"
#include "edata"
#include "evtim"
#include "extst"
#include "fcdesc"
#include "misdat"
#include "mslflt"
#include "rdrsta"
#include "skrsta"
#include "skrdat"
#include "lprnts"
      logical lprnt2,lprnt3
C     --lprnt2 is for LOG file
      equivalence(lprnt2,lprnts(164)), (lprnt,lprnts(203))
      equivalence(lprnt3,lprnts(256))
C  --subroutine argument declarations:
      integer iskr
C  --external declarations:
      character*6 zfmt
      character*2 stri
      character*8 char_skr
      real sepa, dist
      logical lbit, lmatch
      integer rspace
C  --local declarations:
      character*8 skr_name
      integer iseq, tgt_tp, tgt, skr_len
      logical turn_on, no_data, hav_rng
      real xrange, rmtt(3), rel_v(3), tgt_x(3), tgt_v(3), real_dist,
     1 xprj(3), vprj(3), tgt_los(3), dt, apole
      character*4 sides(2)*4
      integer tgt_typ
      data sides(iblue)/'BLUE'/, sides(ired)/'RED '/
C*ENDDEC
      call g_skr_s(iskr)
      skr_name = char_skr(skr_knd)
      skr_len = rspace(skr_name)
      if (skr_faz .eq. skr_on) then
C        --Seeker is already on
         if (lprnt) write(ioutp,6001) skr_name(1:skr_len),
     1    zfmt(mslnr,6),svtm
```

```
         goto 9999
      endif
      if (n_skr_seq .eq. 1) then
         if (lprnt) write(ioutp,6006) skr_name(1:skr_len),
     1    zfmt(mslnr,6)
         goto 500
      else
C        --Don't turn on if missile turns on seekers sequentially
C        --and a seeker later in the sequence is already on
C        --For sequentially operated seekers, turn on if:
C        --1.  Seeker is associated with current sequence step
C        --2.  Seeker is associated with any later sequence steps
         turn_on = .false.
         do 100 iseq = cur_skr_seq,n_skr_seq
            if (lmatch(skr_sequence(1,iseq),mx_skr,iskr)) then
               turn_on = .true.
            endif
100      continue
         if (.not.turn_on) then
C           --seeker not to be turned on
            if (lprnt) write(ioutp,6002) skr_name(1:skr_len),
     1       zfmt(mslnr,6),svtm
            goto 9999
         endif
      endif
500   continue
      if (lbit(mskskr,bit_range_on)) then
C        --This seeker tests on range to target.  Compute
C        --this range using the track that the missile is currently
C        --guiding on.
         call msl_gui_pos(no_data,rmtt,hav_rng,xrange,rel_v)
         if (no_data) then
            write(ioutp,6000) zfmt(mslnr,6), (skr_name(1:skr_len))
            goto 9999
         endif
      elseif (.not.lbit(mskskr,bit_time_on)) then
         call nabort('SKR_TURN_ON...unknown seeker turn_on algorithm')
      endif
      if (skr_knd .eq. irskr) then
         if (lbit(mskskr,bit_range_on)) then
C           --Test is on range.
            call chkrng(iracqr,10.,1.E7,'iracqr...in skr_turn_on')
C           --if hav_rng is false here, use ground truth for xrange
            if(.not.hav_rng)then
               call srgt_tgt(tgt,xprj,vprj,tgt_typ)
               if (tgt .eq. 0) then
                  call nabort('skr_turn_on...Target equal to zero')
               endif
               xrange = dist(xprj,xem)
            endif
            if (xrange .lt. iracqr) then
               skr_faz = skr_on
            else
               if (lprnt) write(ioutp,6004) skr_name(1:skr_len),
     1          zfmt(mslnr,6),svtm,iracqr,xrange
               goto 9999
            endif
         else
            call nabort('SKR_TURN_ON...illegal ir skr turn_on'//
```

```
     1       ' algorithm')
         endif
      elseif (skr_knd .eq. saskr) then
         if (lbit(mskskr,bit_time_on)) then
C           --Test is on time since launch
            if (tym.lt.tymskr) then
               if (lprnt) write(ioutp,6003) skr_name(1:skr_len),
     1          zfmt(mslnr,6),svtm
               goto 9999
            endif
         elseif (lbit(mskskr,bit_range_on)) then
C           --Test is on range to target.
            call chkrng(saacqr,10.,1.E7,'saacqr...in skr_turn_on')
            if (.not.hav_rng) then
                write(ioutp,7000) zfmt(mslnr,6), (skr_name(1:skr_len))
                call nabort('SKR_TURN_ON...no range data for seeker')
            endif
            if (xrange .gt. saacqr) then
               if (lprnt) write(ioutp,6004) skr_name(1:skr_len),
     1          zfmt(mslnr,6),svtm,saacqr,xrange
               goto 9999
            endif
         else
            call nabort('SKR_TURN_ON...illegal sa skr turn_on'//
     1       ' algorithm')
         endif
         call grdrs(illuminator,prant)
         if (.not.lmatch(mslsnc,nmslsn,mslnr)) then
C           --ADD MISSILE TO /RDRSTA/MSLSNC OF LAUNCHER
            nmslsn = nmslsn + 1
            if (nmslsn.gt.mmslsn) then
               call nabort('SKR_TURN_ON...mmslsn exceeded')
            endif
            mslsnc(nmslsn) = mslnr
         endif
         if (lbit(mskmsl,btwsil)) then
C           --Missile requires TWS illumination, which implies that it
C           --will only be on during the last microstep of a fly event
C           --planted from a radar sweep event
            if (lbit(mskmev,bitsnc)) then
               if (last_micro) then
                  skr_faz = skr_on
               endif
            endif
         else
            skr_faz = skr_on
         endif
      elseif (skr_knd .eq. acskr) then
C           --Currently, unclassified active missiles can only
C           --be launched with their seekers on
            call nabort('SKR_TURN_ON...reached classified code')
      elseif (skr_knd .eq. armskr) then
         if (lbit(mskskr,bit_range_on)) then
C           --Test is on range.
            if (xrange .lt. armaqr) then
               skr_faz = skr_on
            endif
         else
            call nabort('SKR_TURN_ON...illegal arm skr turn_on'//
```

```
     1       ' algorithm')
         endif
      else
         call nabort('SKR_TURN_ON...unknown skr_knd = '//stri(skr_knd))
      endif
600   continue
      if (skr_faz .eq. skr_on) then
         if (t_skr_on .eq. xlarge) t_skr_on = svtm
         if (lprnt2 .or. lprnt) then
            if (lbit(mskskr,bit_time_on)) then
               if (lprnt2) write(6,8600) skr_name(1:skr_len),
     1          zfmt(mslnr,6),svtm
               if(lprnt3)then
                  call srgt_tgt(tgt,tgt_x,tgt_v,tgt_tp)
                  write(6,8650)tgt_x(1),tgt_x(2),-tgt_x(3)
                  call prjacc(xe(1,ilaun),ve(1,ilaun),ae(1,ilaun),
     1             svtm-svtime(ilaun),xprj,vprj)
                  write(6,8660)xprj(1),xprj(2),-xprj(3)
                  apole=dist(xprj,tgt_x)
                  write(6,8670)apole
               endif
               if (lprnt) write(ioutp,8500) skr_name(1:skr_len),
     1          zfmt(mslnr,6),svtm
            elseif (lbit(mskskr,bit_range_on)) then
               call srgt_tgt(tgt,tgt_x,tgt_v,tgt_tp)
               real_dist = dist(xem,tgt_x)
               if (lprnt2) then
                  write(6,8100) skr_name(1:skr_len),zfmt(mslnr,6),svtm,
     1             nint(xrange),xrange*ftnmi,nint(real_dist),
     2             real_dist*ftnmi
               endif
               if (lprnt) then
                  write(ioutp,8000) skr_name(1:skr_len),zfmt(mslnr,6),
     1             svtm,nint(xrange),xrange*ftnmi,nint(real_dist),
     2             real_dist*ftnmi
               endif
               if (lprnt3)then
                  write(6,8650)tgt_x(1),tgt_x(2),-tgt_x(3)
                  call prjacc(xe(1,ilaun),ve(1,ilaun),ae(1,ilaun),
     1             svtm-svtime(ilaun),xprj,vprj)
                  write(6,8660)xprj(1),xprj(2),-xprj(3)
                  apole=dist(xprj,tgt_x)
                  write(6,8670)apole
               endif
            else
               call nabort('SKR_TURN_ON...unknown turn_on algorithm'//
     1          ' in mskskr for '//skr_name(1:skr_len)//' seeker')
            endif
         endif
         if (kndaro .eq. aro2) then
C           --Check to see if seeker drops a cover when it turns on
            if (lbit(aero_chg_flg,skr_cvr_off).and.cover_drop) then
               if (lbit(aero_tbl_flg,skr_cvr_off)) then
                  write(ioutp,8700)
               else
                  call sbit(aero_tbl_flg,skr_cvr_off)
                  if (lprnt) write(ioutp,8800) iskr
               endif
            endif
```

```
         else
            continue
         endif
C        --If missile is command guided, dump command guidance data
C        --into seeker's trackbank(s)
         call cg_tk_hndoff(iskr)
      else
         if (lprnt) write(ioutp,6007) skr_name(1:skr_len),
     1    zfmt(mslnr,6),svtm
      endif
9999  return
6000  format('SKR_TURN_ON...Missile ',A,' has an ',A,' seeker, which '/
     1 ' uses a range to target threshold to turn on the seeker.  '/
     2 ' Cannot find any data to drive test - returning')
6001  format('SKR_TURN_ON...',a,' seeker on missile ',a,
     1 ' is already on at svtm = ',f8.3)
6002  format('SKR_TURN_ON...',a,' seeker on missile ',a,
     1 ' is not found in appropriate seeker sequence, so ',/,
     2 ' not turned on at svtm = ',f8.3)
6003  format('SKR_TURN_ON...',a,' seeker on missile ',a,
     1 ' fails time test ,so is not turned on at svtm = ',f8.3)
6004  format('SKR_TURN_ON...',a,' seeker on missile ',a,
     1 ' fails range test ,so is not turned on at svtm = ',f8.3/
     2 ' turn_on range=',F10.2,' range according to track=',F10.2)
6005  format('SKR_TURN_ON...',a,' seeker on missile ',a,
     1 'target is nose-on ,so is not turned on at svtm = ',f8.3)
6006  format('SKR_TURN_ON...',a,' seeker on missile ',a,
     1 ' only one seeker sequence')
6007  format('SKR_TURN_ON...',a,' seeker on missile ',a,
     1 ' NOT turned ON at svtm = ',f8.3)
7000  format('SKR_TURN_ON...Missile ',A,' has an ',A,' seeker, which'/
     1 ' uses a range to target threshold to turn on the seeker.  The '/
     2 ' missile is guiding on a track, but the track has no range'/
     3 ' information in it, so this test cannot be applied')
8000  format('SKR_TURN_ON...',a,' seeker on missile ',a,
     1 ' turns on at svtm = ',f8.3/
     2 ' estimated rng = ',i7,' ft (',f5.1,' nmi),  actual rng = ',
     3 i7,' ft (',f5.1,' nmi)')
8100  format(1X,A,' SEEKER ON MISSILE ',A,
     1 ' TURNS ON AT SVTM = ',F8.3/' ESTIMATED RNG = ',I7,' FT (',
     2 F5.1,' nmi),  ACTUAL RNG = ',I7,' FT (',F5.1,' nmi)')
8500  format('SKR_TURN_ON...',a,' seeker on missile ',a,
     1 ' turns on at svtm = ',f8.3)
8600  format(1X,A,' SEEKER ON MISSILE ',A,' TURNS ON AT SVTM = ',F8.3)
8650  format(t10,'TARGET POSITION: X= ',f10.2,' Y= ',f10.2,
     1       ' Z= ',f10.2)
8660  format(t10,'LAUNCHER POSITION: X= ',f10.2,' Y= ',f10.2,
     1       ' Z= ',f10.2)
8670  format(t10,'APOLE= ',f10.2)
8700  format(' SKR_TURN_ON...WARNING: cover_drop set to true'/
     1 ' Already set aero_tbl_flg bit')
8800  format(' SKR_TURN_ON...seeker cover off for seeker: ',i5)
      end
```

D.2 Listing of Function border.f

This function has multiple entry points.

```
real function border(z,z0)
C*** ----------------------------------------------------------------
C*** THIS SUBROUTINE IS PART OF THE V6.2 VERSION OF BRAWLER.
C*** THIS SUBROUTINE CONTAINS MATERIAL FURNISHED TO THE U.S. GOVERNMENT
C*** WITH UNLIMITED RIGHTS UNDER CONTRACT F49650-89-D5033, BY:
C***    DECISION-SCIENCE APPLICATIONS, INC.
C***    1110 NORTH GLEBE ROAD, SUITE #400,
C***    ARLINGTON, VA 22201.  (703)-243-2500
C*** ----------------------------------------------------------------
C#ABSTRACT GENERALIZED SHAPE FUNCTIONS USED THROUGHOUT BRAWLER
C#PURPOSE  THESE THREE ROUTINES PROVIDE SHAPES USED FOR VALUE FUNCTIONS.
C#AUTHOR   KERCHNER
C#TYPE     MATHEMATICS UTILITY
C#PARAMETER DESCRIPTIONS
CIN  Z      REAL - BASIC INDEPENDENT VARIABLE (SEE FORMULAE BELOW)
CIN  Z0     REAL - WIDTH-DEFINING PARAMETER
COUT BORDER REAL - RANGES FROM 0 FOR Z<<Z0 TO 1 FOR Z>>Z0
COUT CAUCHY REAL - 1 FOR Z NEAR 0; 0 FOR ABS(Z)>>Z0
COUT REWARD REAL - 1 FOR Z NEAR 0; -1 FOR ABS(Z)>>Z0.  A LINEAR
C           TRANSFORMATION OF CAUCHY
C#TECHNICAL DESCRIPTION
C  FORMULAE IN PROGRAM DEFINE EACH FUNCTION.
C  BORDER = SMOOTHED STEP FUNCTION TRANSITIONING AT 0.
C  CAUCHY = BELL-SHAPED CURVE PEAKING AT ORIGIN.
C  REWARD IS CAUCHY BUT WITH RANGE FROM -1 to 1 INSTEAD OF 0 to 1.
C  NOTE RE BORDER:  THE FORMULAE USED FOR POSITIVE AND NEGATIVE ARGUMENTS
ARE
C  DESIGNED TO MAKE BORDER BE CONTINUOUS IN FIRST DERIVATIVE AT Z=0.
C  ALSO, BORDER(Z,Z0) = 1-BORDER(-Z,Z0).
C####
C#AUDIT
C **VERSION V6.2** 12-Oct-95----05:02 DSA -> SURVIAC SITE1 002 SILICONG
C PART-TASK BY Mitchel ON 07-Apr-88 14:17:28 Thu FROM TASK bugfix2
C   Associated with bugfix form 00216 (No change notice)
C MODIFIED BY MITCHEL ON 07-Apr-88  09:46:42 Thu FOR TASK bugfix2
C  Changed expressions of form: z = x*y**2   to : t = y
C                                                 z = x*t*t
C  to simplify expressions and for more efficiency on some machines.
C MODIFIED BY KERCHNER ON 13-Sep-86  08:59:08 Sat FOR TASK bugfix
C  Cosmetics
C#ENTRY POINTS
C  BORDER(Z,Z0)
C  CAUCHY(Z,Z0)
C  REWARD(Z,Z0)
C####
      real u     ,usq   ,z     ,z0 ,t
      REAL cauchy,reward
C*ENDDEC
      if (z.le.0.) then
         t = (z/z0-1.)
         border = 1./(1.+(t*t))
      else
         u = (z/z0+1.)*(z/z0+1.)
         border = u/(1.+u)
      endif
      return
C     ----------------------------------------------------------------
      entry cauchy(z,z0)
      t = (z/z0)
```

```
      border = 1./(1.+(t*t))
      return
C     ------------------------------------------------------------------
      entry reward(z,z0)
      usq = (z/z0)*(z/z0)
      border = (1.-usq)/(1.+usq)
      return
      end
```

D.3 Listing of Suboutine rdrgtk.f

This routine has multiple entry points that use GOTOs to branch back into the main part of the routine. (This routine was not included in the evaluation.)

```
subroutine rdrgtk(jacid,entid,enttyp,exist,trkmod)
C*** ------------------------------------------------------------------
C*** THIS SUBROUTINE IS PART OF THE V6.2 VERSION OF BRAWLER.
C*** THIS SUBROUTINE CONTAINS MATERIAL FURNISHED TO THE U.S. GOVERNMENT
C*** WITH UNLIMITED RIGHTS UNDER CONTRACT F49650-89-D5033, BY:
C***    DECISION-SCIENCE APPLICATIONS, INC.
C***    1110 NORTH GLEBE ROAD, SUITE #400,
C***    ARLINGTON, VA 22201.  (703)-243-2500
C*** ------------------------------------------------------------------
C#ABSTRACT RETRIEVES RADAR TRACK
C#PURPOSE  CALLED WHENEVER IT IS DESIRED TO FETCH A RADAR TRACK
C#AUTHOR   Kramer
C#TYPE     Radar Trackbank Utility
C#PARAMETER DESCRIPTIONS
CIN  JACID  AC-IDX - Radar platform   (ENTRY RDRGTK ONLY)
CIN  ENTID  ENT-ID - Entity ID for which to retrieve track:
C           If enttyp=/par/acent this is an aircraft tail number
C         If enttyp=/par/mslent this is a missile ID number (e.g.  010201)
C           (ENTRY RDRGTK ONLY)
CIN  ENTTYP ENT-TYP - Indicates type of entity defined by entid:
C           /par/acent  => aircraft
C           /par/mslent => missile
C           (ENTRY RDRGTK ONLY)
CIN  TRKMOD LOG - flag indicates mode of retrieval
C           .true. -> TWS track
C           .false.-> scan/STT track
C           (ENTRY RDRGTK ONLY)
COUT EXIST  LOG - .TRUE. if track exists
C           (ENTRY RDRGTK ONLY)
COUT RTKPTR PTR - ptr to /rdrtrk/ for the track to be retrieved
C           (ENTRY RGTKP ONLY)
CIN  TRKIDX RDR-TRK - Track index for track to be fetched. (like
C           /rdrtbk/xrtkid).
C           (ENTRY RDRGTT ONLY)
C#TECHNICAL DESCRIPTION
C
C   IT  IS  ASSUMED  THAT  /RDRSTA/  AND  /RDRTBK/  are  CURRENT  FOR  radar
platform!!
C
C  ENTRIES RDRGTK AND RDRGTT:
C  If a track exists for this entity, then get it from memory and
C  put it in /rdrtrk/.
C  If there is no track yet, then return exist = .FALSE. (ENTRY RDRGTK)
C  or abort (entries rgtkp and rdrgtt).
```

```
C
C  ENTRY RGTKP:
C  If a track exists for this entity, get the pointer.  If there is
C  no track yet, then return exist = .FALSE.
C
C#VARIABLE DESCRIPTIONS
C  ITRK    INT - INDEX OF APPLICABLE TRACK
C####
C#AUDIT
C **VERSION V6.2** 12-Oct-95----04:35 DSA -> SURVIAC SITE1 002 SILICONG
C PART-TASK BY Kramer ON 11-Dec-90 18:22:59 Tue FROM TASK aesa_mods
C MODIFIED BY Eiserman ON 29-Oct-90  20:12:51 Mon FOR TASK aesa_mods
C  Added code to correctly set modmsk for an ESA radar
C PART-TASK BY Olszewski ON 09-Aug-90 14:44:19 Thu FROM TASK bugfix4
C   Associated with bugfix form 90-0204
C MODIFIED BY Olszewski ON 08-Aug-90  15:43:46 Wed FOR TASK bugfix4
C  Changed declarations for testb from integer to logical.
C PART-TASK BY Kramer ON 19-Jul-90 10:01:38 Thu FROM TASK rdr_trackbank
C MODIFIED BY Kramer ON 16-Apr-90  13:22:48 Mon FOR TASK rdr_trackbank
C  Removed code which referenced scan/stt and missile trackbanks.
C  All radar trackbanks have been combined into a single trackbank.
C  Changed call to rdrptk to match new lack of parameters.
C  Also added entry rdrgtt and removed unused entry point rdrgtp.
C PART-TASK BY Kramer ON 28-Oct-88 07:44:27 Fri FROM TASK rdr_see_misl
C MODIFIED BY Kramer ON 21-Oct-88  10:59:36 Fri FOR TASK rdr_see_misl
C  Added ckrngi call on aircraft or missile ID.
C  Added some lprnts.
C PART-TASK BY Kramer ON 01-Oct-88 10:16:07 Sat FROM TASK rdr_see_misl
C MODIFIED BY Kramer ON 12-Sep-88  14:45:20 Mon FOR TASK rdr_see_misl
C  Added code required to consider the new radar trackbanks for missiles.
C  This included adding the new parameter enttyp.
C PART-TASK BY Kramer ON 01-Jun-88 17:28:50 Wed FROM TASK bugfix4
C   Associated with bugfix form 00262 (Change notice 88-56)
C MODIFIED BY Kramer ON 01-Jun-88  11:41:43 Wed FOR TASK bugfix4
C  Added a new entry point that fetches pointer only.  It is named
C  rdrgtp.
C PART-TASK BY Kramer ON 18-Mar-88 16:50:28 Fri FROM TASK antenna
C MODIFIED BY MITCHEL ON 08-Mar-88  16:51:48 Tue FOR TASK antenna
C  Changed reference of TRKFMS to its new name TRKHTS.
C#ENTRY POINTS
C     subroutine rdrgtk(entid,enttyp,exist,trkmod)
C     entry rgtkp(rtkptr)
C     entry rdrgtt(trkidx)
C####
#include "par"
#include "rdrsta"
#include "rdrtbk"
#include "rdrtrk"
#include "lprnts"
      equivalence (lprnts(186),lprnt)
C     --subroutine argument declarations
      integer entid,rtkptr,enttyp,trkidx,jacid
      logical exist,trkmod
C     --external declarations
      character*8 chrid
      character*10 chrent
      character*12 stri
      logical rdr_is_esa
C     --local declarations
```

```
      integer ntrk,trklst(2*mxtkpe),idlst(2*mxtkpe)
      integer modmsk,itrk,ptr,imhere
      logical testb(2*mxtkpe)
C*ENDDEC
50    continue
      ptr = 0
C     --LOOK FOR AN EXISTING TRACK
      if (rdr_is_esa(jacid)) then
         modmsk = 16
      elseif (trkmod) then
         modmsk = 12
      else
         modmsk = 3
      endif
      call havtrk(jacid,entid,enttyp,clrdr,modmsk,exist,ntrk,trklst,
     1 idlst,testb)
C     --WAS A TRACK FOUND?
      if (.not.exist) then
C        --NO TRACK FOUND
         if(lprnt)write(ioutp,1020) chrent(enttyp),chrid(entid,enttyp),
     1    trkmod
         goto 999
      endif
      itrk = trklst(1)
      if (lprnt) write(ioutp,1010) chrent(enttyp),chrid(entid,enttyp),
     1 itrk,trkmod
75    continue
C     --ENTRY RDRGTT JOINS HERE
C     --TRACK FOUND HERE
      if(rtkpts(itrk).eq.0)call nabort('rdrgtk...zero ptr in rtkpts')
      ptr = rtkpts(itrk)
C
100   continue
C     --RGTKP JOINS HERE
      if (ptr.ne.crtrkp) then
C        --DESIRED TRACK IS NOT THE CURRENT ONE
C        --PUT THE CURRENT TRACK
         if (crtrkp.ne.0) call rdrptk
C        --FETCH THE DESIRED TRACK
         call memrd(trkhts,lrdrtk,ptr,0)
         crtrkp = ptr
         if(lprnt)write(ioutp,1030) crtrkp,xrtrk,vrtrk
      else
         if(lprnt)write(ioutp,1040) crtrkp
      endif
999   continue
      return
C    ------------------------------------------------------------------
      entry rgtkp(rtkptr)
      if (rtkptr.eq.0) call nabort('rdrgtk$rgtkp...null ptr')
      ptr = rtkptr
      call match(rtkpts,mrdrtk,imhere,ptr)
      if (imhere .eq. 0 .or. .not.trkvld(imhere)) call nabort(
     1 ' RGTKP...GIVEN PTR DOES NOT REFER TO AN EXISTING TRACK'
     2 //'pointer='//stri(ptr))
      goto 100
C    ------------------------------------------------------------------
      entry rdrgtt(trkidx)
      call ckrngi(trkidx,1,mrdrtk,'trkidx...in rdtgtt')
```

```
      if (.not.trkvld(trkidx)) call nabort('rdrgtt...invalid track '
     1 //stri(trkidx))
      itrk = trkidx
      goto 75
C
1010  format(' RDRGTK...track exists for ',2A,' itrk,trkmod=',
     1 i7,l5)
1020  format(' RDRGTK...track does not exist for ',2A,' trkmod=',l5)
1030  format(' RDRGTK...track fetched: crtrkp,xrtrk,vrtrk: ',
     1 i6,2(/,3E13.5))
1040  format(' RDRGTK...not fetching track, already current for ptr=',
     1 i6)
      end
```