NSSDCA Document B56522-000A

Apollo ARCSAV Level 0a (Raw Cleaned Binary) File Processing Programs and Supplementary Input Files [1]

Program [2]	Description	Main Input Files [3]	Supplementary Input Files	Output Files [4]	Last Revision Date
removenb2nl.f	Remove null bytes up to normal record length	e*.tap or 28c1.*.jbi	dvsfnall.*	asr.*.*, rmnblog.*	2/15/17
divide2ar.f	Divide null-byte removed, selected JBI-format files into ARCSAV records	asr.*	dvsfns.*	asr.*.*, d2rlog.*	2/15/17
divideK2ar.f	Divide JBI-formatted KDM files into ARCSAV records	*.tap.jbi renamed to K*		asr.K*, d2rlog.*K	11/10/16
combinejk15.f	Combine JBI- and KDM-based asr.*.* files for Apollo 15	asr.K*	jkclist15.txt, jkfclog.15	asr.15.*	2/6/17
combinejk17.f	Combine JBI- and KDM-based asr.*.* files for Apollo 17	asr.K*	jkclist17.txt, jkfclog.17	asr.17.*	5/19/17
finalcorr.f	Final correction to asr.*.*	asr.*.*	fcorlist.*	asr.*.*, finalcorrlog.*	5/26/17
finalcorr2.f	2nd final correction to asr.*.*	asr.*.*	ponlysdrlist.txt	asr.*.*, finalcorr2log	4/2/18
finalcorr2r.f	2nd final correction to asr.*.* -- repeat	asr.*.*	ponlysdrlistr.txt	asr.*.*, finalcorr2rlog	4/6/18

[1] There are many supplementary programs, not listed here, that were used primarily for testing at various stages of generating Level 0a files and for correcting specific files. Only the key programs are listed here.

[2] These Fortran (*.f) programs were provided by the data provider, Yosio Nakamura, for documentation purposes only. The programs are provided in alphabetical order on pages 2-15 of this document. The Supplementary Input Files used by these programs are listed in alphabetical order on pages 16-34.

[3] The Main Input files named "e*.tap" are the files that vendor JBI extracted from the ARCSAV tapes. The "28c1.*.jbi" files are those extracted by vendors DBDS and KDM, then converted to the JBI format by the data provider, Yosio Nakamura. These files are archived at the NASA Space Science Data Coordinated Archive (NSSDCA) under these collection identifiers: PSPG-00912 (Apollo 12); PSPG-00913 (Apollo 14); PSPG-00914 (Apollo 15); PSPG-00915 (Apollo 16); and PSPG-0016 (Apollo 17). Please note the vendor files were renamed for these Level 0 collections; the file names are cross-referenced in NSSDCA document B56516, https://nssdc.gsfc.nasa.gov/misc/documents/b56516.pdf.

[4] The Output files named "asr.*.*", "asr.15.*", and "asr.17.*" are grouped into four Level 0a (raw binary, cleaned) data collections by Apollo mission and archived at the NASA Space Science Data Coordinated Archive (NSSDCA) under these collection identifiers: PSPG-00917 (Apollo 12); PSPG-00918 (Apollo 15); PSPG-00919 (Apollo 16); and PSPG-0920 (Apollo 17). The data provider, Yosio Nakamura, determined there were no viable data in the Apollo 14 Level 0 ARCSAV files, and therefore he did not produce Level 0a products. Please note the NSSDCA renamed files contained in the Level 0a collections to "arcsav_ap*_yyyyddd_cleaned" where * is Apollo mission number and yyyyddd is the acquisition year and day. The Level 0 and Level 0a file names are cross-referenced in NSSDCA document B56516, https://nssdc.gsfc.nasa.gov/misc/documents/b56516.pdf. The Output files named *log.* are processing logs and were not archived. Level 0 and 0a data formats are described in NSSDCA document B56515, https://nssdc.gsfc.nasa.gov/misc/documents/b56515.pdf.

combinejk15.f

```
      program combinejk15
c combine JBI & Katalyst files in asrecsc & asrecskb intoasrecsjk
c **6-Feb-17**
      character jfx*5,kfx*4,cfx*6,crj*4
      integer data(1600)
c
      open(2,file='../lists/jkclist15.txt',status='old')
      open(8,file='../logs/jkfclog.15')
10    read(2,'(a5,1xa4,1xa6,i5)',end=19) jfx,kfx,cfx,nrj
      write(crj,'(i4)') nrj
      if (crj(1:1).eq.' ') crj(1:1)='0'
      call system('dd if=../asrecskb/asr.K'//kfx//' bs=6400 skip='
&     //crj//'>aa')
      call system('cat ../asrecsc/asr.'//jfx//' aa'//
&     '>../asrecsjk/asr.'//cfx)
      write(8,1) jfx,kfx,cfx
      write(*,1) jfx,kfx,cfx
1     format(a5,1xa4,' > ',a6)
      go to 10
19    close(2)
      close(8)
      stop
      end
```

combinejk17.f

```
      program combinejk17
c combine JBI & Katalyst files in asrecsc & asrecskb into asrecsjk
c **6-Feb-17**rev. 19-May-17**
      character afx*3,kfx*4,crj*4
      integer data(1600)
c
      open(2,file='../lists/jkclist17.txt',status='old')
      open(8,file='../logs/jkfclog.17')
10    read(2,'(a3,1xa4,i5)',end=19) afx,kfx,nrj
      if (nrj.eq.0) then
       call system('cp -p ../asrecskb/asr.K'//kfx//' ../asr17/asr.17.'//afx)
      else
       write(crj,'(i4)') nrj
       if (crj(1:1).eq.' ') crj(1:1)='0'
       call system('dd if=../asrecskb/asr.K'//kfx//' bs=6400 skip='//crj
&      //'>aa')
       call system('cat ../asr17/save/asr.17.'//afx//' aa'//
&      '>../asr17/asr.17.'//afx)
       end if
      write(8,1) kfx,nrj+1,afx
      write(*,1) kfx,nrj+1,afx
1     format('K',a4,' from record',i5,' added to asr.17.',a3)
      go to 10
19    close(2)
      close(8)
      stop
      end
```

divide2ar.f

```
      program divide2ar
c divide null-byte removed selected JBI-formated files into ARCSAV records
c **12-Aug-16**mod. 21-Sep-16**rev 15-Feb-17**
      character sta*2,day*3,dbf*5,nrec(3)*5
      byte data(30000)
      logical irs
c
      write(*,"('station: '$)")
      read(*,'(a2)') sta
      open(2,file='../lists/dvsfns.'//sta,status='old')
      open(8,file='../logs/d2rlog.'//sta)
10    read(2,'(a3,1xa5)',end=39) day,dbf
      ns=5
      do while (dbf(ns:ns).eq.' ')
       ns=ns-1
       end do
      open(1,file='../asrecsx/asr.'//dbf(1:ns),access='stream',status='old')
      open(7,file='../asrecsa/asr.'//dbf,access='stream')
      write(*,'(a3,1xa5)') day,dbf
      if (day(1:1).eq.'0') then
       ia=3
      else
       ia=2
       end if
      nr=1
      mr=0
20    read(1,end=29) nb
      read(1) (data(i),i=1,nb)
      read(1) nbe
      if (nb.le.60) go to 28
      is=1
      irs=.false.
      do k=1,3
       if (is+7000.lt.nb) then
        if (is+5000.le.nb) then
c search for end of ARCSAV record
         il=nb-7800
         do while (il.gt.is+6400)
          il=il-6400
          end do
         ib=min(il,is+5000)
         nzc=0
         do i=ib,nb
          kd=data(i)
          if (and(kd,'3f'x).eq.0) then
           nzc=nzc+1
          else
           if (nzc.ge.100) go to 25
           nzc=0
           end if
          end do
         ie=nb
         go to 26
25       if (data(i).ne.'15'x) then
          ie=i-ia
         else
```

```
          ie=i-1
          end if
        else
         ie=nb
         end if
       else
        ie=nb
        end if
c output single ARCSAV record
26     nba=ie-is+1
       if (nba.gt.10000.or.(mr.gt.3039.and.(nba.lt.6000)))
&      then
        write(8,"(a5,i5,i6,' skipped')") dbf,mr+1,nba
        go to 28
        end if
       if (nba.gt.0) then
        mr=mr+1
        write(7) nba
        write(7) (data(i),i=is,ie)
        write(7) nba
        if (k.eq.1) mrs=mr
        end if
       if (nba.ne.6400) then
        write(nrec(k),'(i5)') nba
        irs=.true.
       else
        nrec(k)='     '
        end if
       is=ie+1
       end do
      if (irs) write(8,'(a5,i5,3a5)') dbf,mrs,nrec
28    nr=nr+1
      go to 20
29    close(1)
      close(7)
      go to 10
39    close(2)
      close(8)
      stop
      end
```

divideK2ar.f

```
            program divideK2ar
c divide JBI-formated Katalyst files into ARCSAV records
c **31-Oct-16**
      character dbf*5,nrec(3)*5
      byte data(20000)
      logical irs
c
      open(8,file='../logs/d2rlog.K')
      do np=2739,4508
       write(dbf,'(a1,i4)') 'K',np
       open(1,file='../../../katalyst/links/'//dbf,access='stream'
&      ,status='old',err=39)
       open(7,file='../asrecska/asr.'//dbf,access='stream')
       write(*,'(a5)') dbf
       if (np.eq.4430.or.np.eq.4495) then
        ia=3
       else
        ia=2
        end if
       do j=1,2
        read(1) nb
        nb=iswap(nb)
        read(1) (data(i),i=1,nb)
        read(1) nbe
        end do
       mr=0
       n2=0
21     read(1,end=29) nb
       if (nb.eq.0) go to 29
       nb=iswap(nb)
       read(1) (data(i),i=1,nb)
       read(1) nbe
       n2=nb
       if (nb.gt.19000) go to 23
22     data(n2+1)='80'x
       read(1,end=29) nb
       if (nb.eq.0) go to 29
       nb=iswap(nb)
       n1=n2+2
       n2=n1+nb-1
       read(1) (data(i),i=n1,n2)
       read(1) nbe
       if (n2.lt.17432) go to 22
23     is=1
       irs=.false.
       do k=1,3
        if (is+7000.lt.n2) then
         if (is+5000.le.n2) then
c search for end of ARCSAV record
          il=n2-7800
          do while (il.gt.is+6400)
           il=il-6400
           end do
          ib=min(il,is+5000)
          nzc=0
          do i=ib,n2
```

```
           kd=data(i)
           if (and(kd,'3f'x).eq.0) then
            nzc=nzc+1
           else
            if (nzc.ge.100) go to 25
            nzc=0
            end if
           end do
          ie=n2
          go to 26
25        if (data(i).ne.'15'x) then
           ie=i-ia
          else
           ie=i-1
           end if
         else
          ie=n2
          end if
        else
         ie=n2
         end if
c output single ARCSAV record
26      nba=ie-is+1
      if (nba.eq.0) then
       write(*,*) k,mr,is,n1,n2,ie,nb
       stop
       end if
        if (nba.gt.0) then
         mr=mr+1
         write(7) nba
         write(7) (data(i),i=is,ie)
         write(7) nba
         if (k.eq.1) mrs=mr
         end if
        if (nba.ne.6400) then
         write(nrec(k),'(i5)') nba
         irs=.true.
        else
         nrec(k)='     '
         end if
        is=ie+1
        end do
       if (irs) write(8,'(a5,i5,i6,3a5)') dbf,mrs,n2,nrec
       n2=0
       go to 21
29     close(1)
       close(7)
39     end do
      close(8)
      stop
      end

      integer function iswap(in)
c integer swap bytes
      byte ibi(4),ibo(4)
      equivalence (ins,ibi(1)),(io,ibo(1))
      ins=in
      do i=1,4
```

```
 ibo(i)=ibi(5-i)
 end do
iswap=and(io,'7fff'x)
return
end
```

finalcorr.f

```
      program finalcorr
c final correction of asr.* in asr**
c **26-May-17**
      character st*2,day*3
      byte data(6400),kdb
      integer mv(2)
      logical new
      common data,nsb
c
      write(*,"('station: '$)")
      read(*,'(a2)') st
      open(2,file='../lists/fcorlist.'//st, status='old')
      open(8,file='../logs/finalcorrlog.'//st)
      new=.true.
10    read(2,'(a3,i5,o6,z3,i2,2z3)',end=49) day,kr,kb,kv,kc,mv
      if (kr.eq.9999) then
       write(1,rec=lr) data
       close(1)
       new=.true.
       go to 10
       end if
      if (new) then
       open(1,file='../asr'//st//'/asr.'//st//'.'//day,access='direct'
&      ,form='unformatted',recl=6400,status='old')
       write(*,*) day
       new=.false.
       lr=0
       end if
      if (kr.ne.lr) then
       if (lr.ne.0) write(1,rec=lr) data
       read(1,rec=kr) data
       nsb=0
       lr=kr
       end if
c
      kd=data(kb+nsb)
      kd=and(kd,'ff'x)
      if (kd.ne.kv) then
       write(*,'(i4 o6,2z3)') kr,kb,kv,kd
       stop
       end if
      go to (21,22,23,24,25,26,27,28,29) kc
c replace single byte with one supplied
21    data(kb+nsb)=mv(1)
      write(8,2) kr,kb,kv,mv(1)
2     format(i4,o6,':',z3,' >',2z3)
      go to 10
c combine this byte with the one that precedes and backshift one byte
22    lv=data(kb+nsb-1)
      lv=and(lv,'ff'x)
      data(kb+nsb)=or(lv,kv)
      write(8,3) kr,kb+nsb-1,lv,kv,data(kb+nsb)
3     format(i4,o6,':',2z3,' >',z3)
      call bshift(kb+nsb)
      go to 10
c combint this byte with the one that follows and backshift one byte
```

```
23    nv=data(kb+nsb+1)
      nv=and(nv,'ff'x)
      data(kb+nsb)=or(kv,nv)
      write(8,3) kr,kb+nsb,kv,nv,data(kb+nsb)
      call bshift(kb+nsb+1)
      go to 10
c remove this byte and backshift one byte
24    write(8,4) kr,kb+nsb,kv
4     format(i4,o6,':',z3,' removed')
      call bshift(kb+nsb)
      go to 10
c split a byte into two as given and forward shift one byte
25    call fshift(kb+nsb)
      data(kb+nsb)=mv(1)
      data(kb+nsb+1)=mv(2)
      write(8,2) kr,kb+nsb,kv,mv
      nsb=nsb+1
      go to 10
c insert the given byte before this byte and foward shift one byte
26    if (mv(2).eq.0) read(2,'(i10)') mv(2)
      do n=1,mv(2)
       call fshift(kb+nsb)
       data(kb+nsb)=mv(1)
       write(8,5) kr,kb+nsb,mv(1),kv
5      format(i4,o6,': insert',z3,' before',z3)
       nsb=nsb+1
       end do
      go to 10
c correction uncertain – flag it with msb set
27    data(kb+nsb)=or(kv,'80'x)
      write(8,2) kr,kb+nsb,kv,data(kb+nsb)
      go to 10
c remove given number of preceeding bytes
28    write(8,"(i4,o6,':',i4,' preceeding bytes removed')") kr,kb+nsb,mv(1)
      do i=1,mv(1)
       call bshift(kb+nsb-1)
       end do
      go to 10
c set msb for the given range and remove or insert 80 for a given number of
bytes
29    read(2,'(i10,o6)') nbx,kbe
      mb1=kb+nsb
      mb2=kbe+nsb
      do i=mb1,mb2
       kd=data(i)
       data(i)=or(kd,'80'x)
       end do
      write(8,"(i4,o6,' to',o6,': set msb')") kr,mb1,mb2
      if (nbx.lt.0) then
       do i=1,-nbx
        is=mb2-i+1
        kdb=data(is)
        call bshift(is)
        write(8,4) kr,is,kdb
        end do
      else if (nbx.gt.0) then
       do i=1,nbx
        is=mb2+i
        kd=data(is)
        call fshift(is)
        data(is)='80'x
        write(8,5) kr,is,'80'x,kd
        nsb=nsb+1
        end do
       end if
      go to 10
c
49    close(2)
      stop
      end

      subroutine bshift(ns)
c backward shift the remainder by a byte
      byte data(6400)
      common data,nsb
c
      do i=ns,6399
       data(i)=data(i+1)
       end do
      nsb=nsb-1
      return
      end

      subroutine fshift(ns)
c forward shift the remainder by a byte
      byte data(6400)
      common data
c
      do i=6399,ns,-1
       data(i+1)=data(i)
       end do
      return
      end
```

finalcorr2.f

```
      program finalcorr2
c 2nd final correction of asr.* in asr**
c **2-Apr-18**
      character st*2,day*3,inf*34,outf*30
      byte data(6400),kdb
      integer mv(2)
      logical ep
c
      open(2,file='../lists/ponlysdrlist.txt', status='old')
      open(8,file='../logs/finalcorr2log')
      read(2,*) ks,kd,kr
10    write(st,'(i2)') ks
      write(day,'(i3)') kd
      if (day(1:1).eq.' ') day(1:1)='0'
      inf='../asr'//st//'/asr.'//st//'.'//day
      outf='../asrcor2/asr.'//st//'.'//day
      open(1,file=inf,access='direct',form='unformatted'
&     ,recl=6400,status='old')
      call system('cp '//inf//' '//outf)
      call system('chmod 644 '//outf)
      open(7,file=outf,access='direct',form='unformatted'
&     ,recl=6400,status='old')
      write(*,*) st//'.'//day
c
20    read(1,rec=kr) data
      do k=1,6400
       kv=data(k)
       ep=.true.
       if (and(kv,'80'x).eq.0) then
        iv=kv
        do i=1,7
         if (and(iv,1).eq.1) ep=.not.ep
         iv=rshift(iv,1)
         end do
        if (ep) then
         data(k)=or(kv,'80'x)
         write(7,rec=kr) data
         write(8,"(a6,2i5,z3,' >',z3)") st//'.'//day,kr,k,kv,data(k)
         end if
        end if
       end do
c
      ld=kd
      read(2,*,end=49) ks,kd,kr
      if (kd.ne.ld) then
       close(7)
       go to 10
       end if
      go to 20
c
49    close(2)
      close(7)
      close(8)
      stop
      end
```

finalcorr2r.f

```
      program finalcorr2r
c 2nd final correction of asr.* in asr** - repeat
c **6-Apr-18**
      character st*2,day*3,inf*34,outf*30
      byte data(6400),kdb
      integer mv(2)
      logical ep
c
      open(2,file='../lists/ponlysdrlistr.txt', status='old')
      open(8,file='../logs/finalcorr2rlog')
      read(2,*) ks,kd,kr
10    write(st,'(i2)') ks
      write(day,'(i3)') kd
      if (day(1:1).eq.' ') day(1:1)='0'
      inf='../asr'//st//'/asr.'//st//'.'//day
      outf='../asrcor2/asr.'//st//'.'//day
      open(1,file=inf,access='direct',form='unformatted'
&     ,recl=6400,status='old')
      call system('cp '//inf//' '//outf)
      call system('chmod 644 '//outf)
      open(7,file=outf,access='direct',form='unformatted'
&     ,recl=6400,status='old')
      write(*,*) st//'.'//day
c
20    read(1,rec=kr) data
      do k=1,6400
       kv=data(k)
       ep=.true.
       if (and(kv,'80'x).eq.0) then
        iv=kv
        do i=1,7
         if (and(iv,1).eq.1) ep=.not.ep
         iv=rshift(iv,1)
         end do
        if (ep) then
         data(k)=or(kv,'80'x)
         write(7,rec=kr) data
         write(8,"(a6,2i5,z3,' >',z3)") st//'.'//day,kr,k,kv,data(k)
         end if
        end if
       end do
c
      ld=kd
      read(2,*,end=49) ks,kd,kr
      if (kd.ne.ld) then
       close(7)
       go to 10
       end if
      go to 20
c
49    close(2)
      close(7)
      close(8)
      stop
      end
```

removenb2nl.f

```
      program removenb2nl
c remove null bytes up to normal record length
c **11-Aug-16** rev 19-Sep-16** for both JBI & DBDS files **rev 15-Feb-17
      character sta*2,flx*5,jbi*32,dbk*30
      byte data(30000)
      logical eof
      common data,nb
      data jbi/'../../jbi2/link2taps/e0040100342'/
&     ,dbk/'../../../databank/in_jbi/28c1.'/
c
      write(*,"('station: '$)")
      read(*,'(a2)') sta
      open(2,file='../lists/dvsfnall.'//sta,status='old')
      open(8,file='../logs/rmnblog.'//sta)
10    read(2,'(4xa5)',end=39) flx
      if (flx(5:5).ne.' '.and.flx(5:5).ne.'r') then
c JBI-read file
       open(1,file=jbi//flx//'.tap',access='stream',status='old')
      else
c DBDS-read files
       ns=5
       do while (flx(ns:ns).eq.' ')
        ns=ns-1
        end do
       open(1,file=dbk//flx(1:ns)//'.jbi',access='stream',status='old')
       end if
      open(7,file='..//asrecsx/asr.'//flx,access='stream')
      write(*,*) flx
      nr=1
      eof=.false.
20    read(1,end=29) nb
      if (nb.eq.0) go to 20
      nb=iswap(nb)
      if (nb.eq.1.and.eof)  go to 29
      eof=nb.eq.1
      if (nb.gt.30000) then
       write(*,*) nr,nb
       if (nr.lt.1060) stop
       go to 29
       end if
      read(1) (data(i),i=1,nb)
      read(1) nbe
      if (eof) go to 20
      if (nb.gt.19200) then
       nbs=nb
       i=1
       do while (i.le.nb.and.nb.gt.19200)
        if (data(i).eq.0) then
         call bshift(i)
        else
         i=i+1
         end if
        end do
       if (nb.ne.nbs) write(8,1) flx,nr,nbs,nb
1      format(a5,i5,':',i6,' >',i6)
       end if
      write(7) nb
      write(7) (data(i),i=1,nb)
      write(7) nb
      nr=nr+1
      go to 20
29    close(1)
      close(7)
      go to 10
39    close(2)
      close(8)
      stop
      end

      integer function iswap(in)
c integer swap bytes
      byte ibi(4),ibo(4)
      equivalence (ins,ibi(1)),(io,ibo(1))
      ins=in
      do i=1,4
       ibo(i)=ibi(5-i)
       end do
      iswap=and(io,'7fff'x)
      return
      end

      subroutine bshift(ns)
c backward shift the remainder by a byte
      byte data(30000)
      common data,nb
c
      nb=nb-1
      do i=ns,nb
       data(i)=data(i+1)
       end do
      return
      end
```

dvsfnall.15

```
 92 2253b
 93 6416a
 94 60270
 94 1
 95 60727
 96 61432
 96 2
 97 64ba2
 98 5f8f4
 99 6515a
100 6343c
101 61fd6
102 6327f
103 5e663
104 61aac
105 22ea9
106 62796
107 5e7b9
108 60ba5
109 5f117
110 5f216
111 63cb4
112 5ed70
113 5ebb8
114 5e0bd
115 5f31e
116 6311c
117 5ef61
118 5ded2
119 62360
120 64e4a
121 5ee02
121 309
122 233
123 234
124 235
125 236
126 237
127 238
128 239
129 242
130 240
130 240r1
131 241
132 156
132 156r1
133 157
133 157r1
134 158
135 159
136 160
137 161
138 162
139 163
140 164
141 165
```

```
142 166
143 167
144 168
145 169
146 170
147 171
148 172
149 173
150 174
151 175
152 176
153 177
154 178
155 179
156 180
157 181
158 182
159 183
160 184
161 185
162 263
163 264
164 265
165 266
165 266r1
166 267
167 268
168 269
169 270
170 271
171 272
172 283
173 284
174 285
175 286
176 287
177 288
178 289
179 290
180 291
181 292
```

dvsfnall.17

```
092 601d1
093 65147
094 63045
095 60917
096 5dbb6
097 5f83b
098 63355
099 5e810
100 307
101 6134d
102 623e7
103 312
108 634bf
109 5da2f
110 5dc22
111 5e3a3
112 63d87
113 650e5
113 305
114 5f561
114 303
115 63273
115 306
116 61e40
121 6483b
122 63a43
123 23b5a
124 5fcb4
124 304
125 6273f
126 308
127 622c2
128 5ec73
129 650dc
129 310
130 5fdf8
130 311
131 62fb3
132 243
136 244
137 245
138 246
139 247
140 248
141 249
142 250
143 251
144 252
145 196
146 197
147 198
148 199
149 200
150 201
151 202
```

152	293
153	294
154	295
155	296
156	297
157	298
158	299
159	300
160	301
161	302
162	76
163	77
163	77r1
164	78
165	79
166	80
167	81
168	82
169	83
170	84
171	85
171	85r1
172	213
173	214
174	215
175	216
176	217
177	218
178	219
179	220
180	221
181	222

dvsfns.15

```
 92 2253b
 93 6416a
 94 1
 95 60727
 96 61432
 97 64ba2
 98 5f8f4
 99 6515a
100 6343c
101 61fd6
102 6327f
103 5e663
104 61aac
105 22ea9
106 62796
107 5e7b9
108 60ba5
109 5f117
110 5f216
111 63cb4
112 5ed70
113 5ebb8
114 5e0bd
115 5f31e
116 6311c
117 5ef61
118 5ded2
119 62360
120 64e4a
121 309
122 233
123 234
124 235
125 236
126 237
127 238
128 239
129 242
130 240r1
131 241
132 156r1
133 157r1
134 158
135 159
136 160
137 161
138 162
139 163
140 164
141 165
142 166
143 167
144 168
145 169
146 170
147 171
```

```
148 172
149 173
150 174
151 175
152 176
153 177
154 178
155 179
156 180
157 181
158 182
159 183
160 184
161 185
162 263
163 264
164 265
165 266
166 267
167 268
168 269
169 270
170 271
171 272
172 283
173 284
174 285
175 286
176 287
177 288
178 289
179 290
180 291
181 292
```

dvsfns.17

```
092 601d1
093 65147
094 63045
095 60917
096 5dbb6
097 5f83b
098 63355
099 5e810
100 307
101 6134d
102 623e7
103 312
108 634bf
109 5da2f
110 5dc22
111 5e3a3
112 63d87
113 650e5
114 303
115 63273
116 61e40
121 6483b
122 63a43
123 23b5a
124 5fcb4
125 6273f
126 308
127 622c2
128 5ec73
129 650dc
130 311
131 62fb3
132 243
136 244
137 245
138 246
139 247
140 248
141 249
142 250
143 251
144 252
145 196
146 197
147 198
148 199
149 200
150 201
151 202
152 293
153 294
154 295
155 296
156 297
157 298
158 299
```

```
159 300
160 301
161 302
162 76
163 77r1
164 78
165 79
166 80
167 81
168 82
169 83
170 84
171 85r1
172 213
173 214
174 215
175 216
176 217
177 218
178 219
179 220
180 221
181 222
```

fcorlist.17

```
098 2338 10453 1f 6 40  2
098 2338 10455 61 6 40  2
098 2338 10457 a2 6 40  2
098 2590  2671 40 9
       6  2742
098 2591  5140 08 6 80  0
     367
    9999
111  288 13443 40 6 7f  1
111  288 13443 40 6 7c  1
    9999
113 2293  2453 40 6 40  1
    9999
116   61   550 7f 6 12  1
116  107 13540 7f 6 7c  1
116  107 13545 7f 6 80  1
116  108 12006 7f 6 40  1
116  108 12075 7f 6 40  1
116  108 12344 7f 6 40  1
116  108 12443 7f 6 40  1
116  108 12534 7f 6 7c  1
116  108 13143 7f 6 7c  1
116  137 11016 61 6 40  1
116  166 11236 7f 6 40  1
116  168 11246 7f 6 40  1
116  168 11277 7f 6 40  1
116  168 11404 7f 6 40  1
116  168 11573 7f 6 40  1
116  239 11636 7f 6 40  1
116  255 11606 7f 6 40  1
116  352 12406 7f 6 40  1
116  440 12124 61 6 40  1
116  597 12724 61 6 40  1
116  597 13047 7f 6 7c  1
116  670 12524 61 6 40  1
116  693 13752 40 6 7c  1
116  843 11676 7f 6 40  1
116  872 12040 7f 6 7c  1
116 1027 13046 7f 6 40  1
116 1107  6233 40 6 40  1
116 1107  7652 40 4
116 1119  7654 61 6 40  1
116 1130 12276 7f 6 40  1
116 1156 13206 7f 6 40  1
116 1160 11140 7f 6 7c  1
116 1203 13176 7f 6 40  1
116 1331 13536 7f 6 40  1
116 1346 10772 67 5 64 43
116 1346 12005 7f 6 40  1
116 1346 12174 7f 6 40  1
116 1493 11022 7f 5 e0 1f
116 1675 13046 7f 6 40  1
    9999
124 2277  6062 20 6 7c  1
124 2530    13 07 1 67
124 2530    14 01 1 61
```

```
124 2557  3162 70 6 80  2
124 2806  2511 40 6 40  1
124 3122  5111 04 1 85
124 3122  5112 01 4
    9999
146 2972 10600 04 6 40  1
146 2972 10600 04 6 20  1
146 2998  1467 1f 6 40  2
    9999
147 1089 10375 0b 6 80  2
147 2621  1221 40 9
       0  1330
147 2938 10771 43 6 38  1
147 2938 10771 43 6 64  1
    9999
161 2983 13765 40 4
    9999
163  421  2031 43 6 38  1
163  421  2031 43 6 64  1
    9999
165 2185 12231 5e 9
       2 12326
165 2185 13767 45 4
165 2185 13770 68 4
    9999
177  108 12605 4c 6 40  2
177  108 13767 07 4
177  108 13770 68 4
    9999
```

jkclist15.txt

```
6416a 4430 15.093 1983
6327f 4439 15.102 1641
61aac 4441 15.104 1410
22ea9 4442 15.105 2622
5e7b9 4444 15.107 2607
5f117 4446 15.109 1401
5f216 4447 15.110 1368
5e0bd 4451 15.114  894
5ded2 4455 15.118 1710
```

jkclist17.txt

```
098 4495 2580
110 4503 2970
113 4506  814
115 4508  918
124 2739 1218
129 2744 1287
171 4598    0
```

jkfclog.15

```
6416a 4430 > 15.093
6327f 4439 > 15.102
61aac 4441 > 15.104
22ea9 4442 > 15.105
5e7b9 4444 > 15.107
5f117 4446 > 15.109
5f216 4447 > 15.110
5e0bd 4451 > 15.114
5ded2 4455 > 15.118
```

jkfclog.17

```
K4495 from record 2581 added to asr.17.098
K4503 from record 2971 added to asr.17.110
K4506 from record  815 added to asr.17.113
K4508 from record  919 added to asr.17.115
K2739 from record 1219 added to asr.17.124
K2744 from record 1288 added to asr.17.129
K4598 from record    1 added to asr.17.171
```

ponlysdrlist.txt

```
12    96    42
12    111   1793
12    114   1631
12    132   1717
12    135   360
12    135   649
12    135   842
12    135   994
12    135   1000
12    135   1284
12    135   1287
12    135   1639
12    135   2145
12    135   2168
12    135   2410
12    135   2608
12    135   2850
12    135   2916
12    135   3090
12    139   528
12    139   996
12    139   2035
12    139   2450
12    139   2595
12    141   2794
12    142   136
12    142   161
12    142   582
12    142   734
12    142   840
12    142   1163
12    142   1212
12    142   1303
12    142   1393
12    142   1645
12    142   1685
12    142   2351
12    143   163
12    143   449
12    143   564
12    143   1227
12    143   1595
12    143   1601
12    143   1887
12    143   2182
12    143   2541
12    143   2721
12    147   971
12    147   2624
12    147   2751
12    147   3126
12    148   1677
12    148   1814
12    148   2271
12    148   2329
12    148   2378
```

12	148	3032
12	148	3044
12	152	2938
12	155	218
12	181	14
12	181	20
15	105	2749
15	117	53
15	135	2742
16	99	1145
16	106	3175
16	145	35
16	155	2785
16	164	240
16	165	112
17	94	2148
17	99	9
17	103	228
17	103	357
17	103	1016
17	103	1164
17	111	1133
17	114	2411
17	115	52
17	116	61
17	116	418
17	130	228
17	130	399
17	130	585
17	130	1519
17	131	3002
17	147	309
17	147	423
17	164	10
17	164	580
17	164	586
17	164	596
17	164	598
17	164	660
17	164	678
17	164	696
17	164	712
17	164	746
17	164	838
17	164	856
17	164	876
17	164	900
17	164	902
17	164	922
17	164	1052
17	164	1142
17	164	1156
17	164	1176
17	164	1220
17	164	1222
17	164	1236
17	164	1240
17	164	1302

17	164	1370
17	164	1374
17	164	1416
17	164	1448
17	164	1450
17	164	1460
17	164	1466
17	164	1476
17	164	1503
17	164	1507
17	164	1536
17	164	1556
17	164	1562
17	164	1576
17	164	1659
17	164	1716
17	164	1718
17	164	1755
17	164	1765
17	164	1780
17	164	1786
17	164	1798
17	164	1800
17	164	1876
17	164	1911
17	164	1971
17	164	1975
17	164	2089
17	164	2178
17	164	2260
17	164	2317
17	164	2336
17	164	2545
17	164	2583
17	164	2675
17	164	2730
17	164	2745
17	164	2749
17	164	2766
17	164	2772
17	164	2776
17	164	2786
17	164	2791
17	164	2792
17	164	2802
17	164	2813
17	164	2815
17	164	2818
17	164	2822
17	164	2835
17	164	2877
17	164	3005
17	164	3006
17	164	3007
17	164	3008
17	164	3009
17	164	3012
17	164	3014

```
17    164   3028
17    164   3029
17    164   3031
17    164   3036
17    164   3046
17    164   3049
17    164   3072
17    164   3082
17    164   3089
17    164   3099
17    164   3102
17    164   3108
17    164   3111
17    164   3127
17    164   3135
17    164   3136
17    164   3143
17    164   3150
17    164   3162
17    164   3165
17    164   3168
17    164   3171
17    164   3172
17    164   3175
17    164   3176
17    164   3177
17    164   3178
```

ponlysdrlistr.txt

```
12    181   14
12    181   20
15    105   2749
16    99    1145
```