WM-1
PDR
Return to
WM
623-SS

JUL 1 2 1982

Distribution:
WMHL file
WMHL r/f
WM r/f
NMSS r/f
JBMartin
REBrowning
MJBell
SASilling & r/f
HJMiller
JOBunting
MJWise
JShields, ADM
PDR

WMHL: ~~3104.2~~
109.9

MEMORANDUM FOR: Malcolm R. Knapp
High-Level Waste Licensing
Management Branch
Division of Waste Management

FROM: Stewart A. Silling
High-Level Waste Licensing
Management Branch
Division of Waste Management

SUBJECT: FILE MAINTENANCE AT BNL

The attached report documents the BNL computer file maintenance procedures which I have found to be the most effective. Note that the procedures call for one individual, the "key user," to act as caretaker for everyone's files. I recommend that you designate such an individual.

ORIGINAL SIGNED BY
Stewart A. Silling
High-Level Waste Licensing
Management Branch
Division of Waste Management

Enclosure:
As stated

8208040034 820712
PDR WASTE
WM-1 PDR

OFFICE	WMHL SAS						
SURNAME	SASilling:lmc						
DATE	7/12/82						

NRC FORM 318 (10/80) NRCM 0240 OFFICIAL RECORD COPY ☆ USGPO: 1980—329-82

3104.2/SAS/82/06/30/0

WMHL FILE MANAGEMENT AT THE BROOKHAVEN COMPUTER SYSTEM

1. Introduction

The increased use of the Brookhaven computer system by several members of this branch has made it desirable to define procedures for maintaining permfiles in an orderly way. The goals of these procedures are as follows:

1) To assure that users have immediate access to current files.
2) To assure that old files can be accessed if necessary.
3) To avoid excessive cost to the NRC or interference with other users.

2. Key user

The branch should identify one person as the key user for the group. The key user is responsible for performing the file maintenance tasks described below.

3. Management of MFZ permfiles

3.1 Creation of files

ADPS requests that all MFZ permfiles be cataloged under ID=ZZRNRC. Complying with this request is advantageous to us because ADPS periodically backs up all ZZRNRC files on tape.

ADPS also requests that any permfile name start with the last name of the person who cataloged it (e.g., SILLINGSWIFTPTC). It is a good idea to have the key user's name appear on a permfile containing a widely used program.

There is no problem with cataloging files with an infinite retention period as long as the key user periodically discards files not being used (see below).

An except-read (XR) password should be used to protect all important programs. (See SCOPE 2.1 Reference Manual, pages 6-11 and 6-15.)

3.2 File structure

ADPS recommends that programs stored on MFZ permfiles consist of three partitions. (A partition is a kind of end-of-file marker.)

(1) OLDPL (program source (FORTRAN) stored in a form used by UPDATE, allowing the user to make small changes easily)

(2) LGO (relocatable binary that comes out of the compiler)

(3) ABS (nonrelocatable binary that comes out of the loader)

Any deviation from this format should be documented clearly.

3.3 Backups

ADPS periodically backs up on tape all ZZRNRC files on MFZ. Any such file that has been accidently destroyed can be restored by calling Emily Robinson or Jim Shields. However, since ADPS does not follow a fixed schedule in backing up files, it is a good idea to check with them before purging any file that cannot be easily recreated and has not been on the system a long time.

3.4 Housecleaning

To avoid keeping useless files on the system, the key user should obtain an AUDIT of all ZZRNRC permfiles on a monthly basis. (See SCOPE 2.1 Reference Manual, page 6-49.) He should purge any file belonging to WMHL which is no longer useful (including old cycle numbers of current files) after verifying with ADPS that any important data has been backed up, and after checking with the owner.

The PURGE control card for SCOPE 2.1 looks like this:

```
PURGE,X,pfname,ID=ZZRNRC[,PW=password][,CY=cycle].
```

Unlike the NOS/BE PURGE command, the first parameter (X) is a dummy which will <u>not</u> become a local file containing the purged permfile.

ADPS recommends keeping infrequently used MFZ programs on tape (see NRC Computer User's Guide, page III-14), but the extra control cards and the possibility of parity errors makes this inconvenient in many cases.

4. Management of MFA permfiles

4.1 Backing up EDITOR files

Contrary to some people's claims, MFA is not particularly unreliable most of the time. However, the despair resulting from loss of a lengthy EDITOR file when the system crashes justifies precautionary measures.

Any EDITOR file should be backed up as a permfile every 10 minutes and immediately upon BATCHing an input file to MFZ.

In order to avoid keeping too many versions of the file, normally only the two most recent versions should be kept.

The following example shows how to back up EDITOR files with minimum effort:

```
..attach,a,dvmsample3,id=user.                (Attach permfile.)
PF CYCLE=004
..e,a,s                                       (Edit the file.)
.
.
.
Make changes for 10 minutes
.
.
.

..s,b,n,o                                     (Save changes.)
..catalog,b,dvmsample3,id=user,rp=45.         (Make a new cycle.)
CT ID=USER          PFN=DVMSAMPLE3
CT CY=005           000640 WORDS              (New cycle is 5.)

..purge,s,dvmsample3,id=user,cy=3.            (Keep only last 2
PR ID=USER          PFN=DVMSAMPLE3            cycles.)
PR CY=003           000576 WORDS
..return,s.
.
.
.
Make more changes for 10 minutes
.
```

```
.
.

..s,c,n,o                                    (Save changes.)
..catalog,c,dvmsample3,id=user,rp=45.        (Make a new cycle.)
NEWCYCLE CATALOG
CT ID=USER          PFN=DVMSAMPLE3
CT CY=006           000704 WORDS             (New cycle is 6.)
..discard,a.                                 (Get rid of cycle 4.)
.
.
.
Make more changes for 10 minutes.
.
.
.

..s,d,n,o                                    (Save changes.)
..catalog,d,dvmsample3,id=user,rp=45.        (Make a new cycle.)
NEWCYCLE CATALOG
CT ID=USER          PFN=DVMSAMPLE3
CY CY=007           000704 WORDS             (New cycle is 7.)
..discard,b.                                 (Get rid of cycle 5.)
```

(This process can go on indefinitely. Every 10 minutes, the file is saved as the next higher letter in the alphabet and cataloged. The second to the last cycle is then discarded.)

4.2 Tape Backups

The key user should back up all permfiles owned by WMHL staff on MFA weekly. Occasionally (e.g., every 4 months) a tape containing all files backed up over the previous period should be stored permanently and not written on ever again.

A procedure for backing up files belonging to WMHL staff has been written. It is invoked by submitting the following job:

```
USER,STMFA,NT2,T777.
ACCOUNT,USER,1349.
ATTACH,X,ARC,ID=SAS.
COPY,X,ARC.
```

```
RETURN,X.
ARC,OLD=oldvsn,NEW=newvsn.
```

In the above job, oldvsn is the tape number (starting with the letter K) of the previous backup tape whose contents will be merged with any new permfiles stored since the last backup. If no merge of previously backed up files is desired, use OLD=Ø (that's zero). The parameter newvsn is the tape number of the tape to be written. Tapes can be reserved by calling the Philips Building computer operator at x27713 and asking for a Brookhaven K-tape.

In order to avoid using excessive tapes, a pair of tapes can be used for weekly backups. On alternate weeks the tapes swap roles as new and old tapes. (See Figure 1.)

About every 4 months an "untouchable" tape should be written containing all files up to that point. This is done just like writing a weekly tape. The OLD tape would be the most recent weekly tape, and the NEW tape would be a new one requested from the Philips Building operator. The untouchable tape should never be written on again, and should be used only to recover specific files. At the time the untouchable tape is written, a weekly tape should also be written.

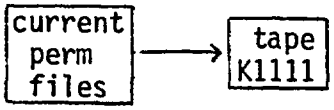
If there are so many files that there is not enough room on the tape, a new sequence should be started using the OLD=Ø parameter. This will prevent merging of an old tape. When this is done, the most recent weekly tape should thereafter be regarded as untouchable.

A listing of the ARC procedure is given in Apendix A. The key user should update the AUDIT commands to reflect any change in personnel. A sample ARC alphabetical file listing is given in Appendix B.

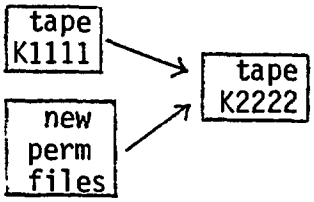
4.3 Recovering backed up files

The ARC program described above sets up temporary permfiles under ID=WMHL which are then written to tape by the sytem utility DUMPF. The names of the backed up files consist of the original permfile ID, the original permfile name, the last date of alteration, and the cycle number. For example, if a user has a permfile called DVMSAMPLE, under ID=SAS, cycle 7, which was last changed on 11/6/81, the backed up file would be SASDVMSAMPLE11Ø681C7,ID=WMHL.

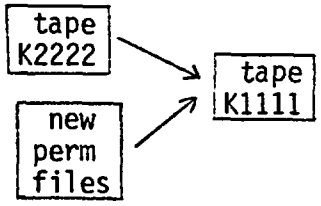
First week: ARC,OLD=Ø,NEW=K1111.



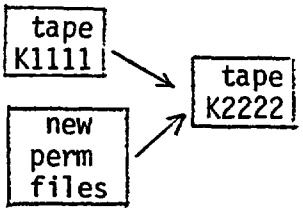
Second week: ARC,OLD=K1111,NEW=K2222.



Third week: ARC,OLD=K2222,NEW=K1111.



Fourth week: ARC,OLD=K1111,NEW=K2222.



.
.
.
etc.

Figure 1. Example of swapping backup tapes on alternate weeks.

Every time a backup tape is written, an alphabetical listing of its contents appears in the printout. These printouts must be kept by the key user.

To recover a backed up file, submit a job like the following:

```
USER,STMFA,NT1,T17,P3.
ACCOUNT,USER,1349.
LABEL,DUMTAPE,D=PE,R,F=S,VSN=KØØØ3.
LOADPF,I=INPUT.
*EOR
PF=SASDVMSAMPLE11Ø681C7,ID=WMHL.
```

Of course, the tape number and permfile names would be different.

4.4 Housecleaning

All users should be urged not to catalog MFA permfiles for longer than 45 days unless necessary.

The key user should obtain a monthly AUDIT of MFA permfiles belonging to the group and nag the owners of unused permfiles to purge them.

Appendix A. ARC Listing

```
REWIND,LGO.
FTN,I=WIPE,L=0,GO.
WIPCCL.
COMMENT.
COMMENT. CATALOG UNDER ID=WMHL ALL FILES TO BE BACKED UP
COMMENT.
AUDIT,LF=CATAUD,ID=KENAN.
AUDIT,LF=CATAUD,ID=KNAPP.
AUDIT,LF=CATAUD,ID=LLL.
AUDIT,LF=CATAUD,ID=ORNSTEIN.
AUDIT,LF=CATAUD,ID=QUINN.
AUDIT,LF=CATAUD,ID=SAS.
AUDIT,LF=CATAUD,ID=WISE.
REWIND,LGO.
FTN,I=CAT,L=0,GO.
CATCCL.
COMMENT.
COMMENT. MERGE THE OLD TAPE.
COMMENT.
IFE,NUM(OLD)=0,NOMERGE.
LABEL,DUMTAPE,D=PE,R,F=S,VSN=OLD.
LOADPF.
RETURN,DUMTAPE.
ENDIF,NOMERGE.
COMMENT.
COMMENT. WRITE THE NEW TAPE
COMMENT.
REQUEST,DUMTAPE,PE,3,N,RING,VSN=NEW.
DUMPF,PW=MODE1,ID=WMHL.
COMMENT.
COMMENT. CREATE ALPHABETICAL LIST OF FILES ON NEW TAPE
```

```
COMMENT.
AUDIT,LF=LIST,ID=WMHL.
REWIND,LGO.
FTN,I=LIST,L=0,GO.
COMMENT.
COMMENT. PURGE ALL FILES UNDER ID=WMHL
COMMENT.
REWIND,WIPAUD.
AUDIT,LF=WIPAUD,ID=WMHL.
REWIND,WIPE.
REWIND,LGO.
FTN,I=WIPE,L=0,GO.
WIPCCL.
REVERT.
.DATA,WIPE.
      PROGRAM WIPE(WIPAUD,WIPCCL,
     *   TAPE1=WIPAUD,TAPE2=WIPCCL,
     *   OUTPUT,TAPE6=OUTPUT)
C
C PURPOSE -- GENERATE CCL PROCEDURE TO PURGE ALL FILES UNDER
C             ID=WMHL
C
      IMPLICIT INTEGER(A-Z)
C
      DIMENSION A(12)
      DIMENSION ACCOUNT(5)
      DIMENSION PFN(4)
C
C         SET ACCOUNT NUMBERS TO BACK UP
      DATA NAC /1/
      DATA ACCOUNT /'1349', 4*' '/
C
C         SET NEW ID
      DATA NEWID /'WMHL'/
C
C         WRITE PROCEDURE HEADER
```

```
      REWIND 2
      WRITE(2,2000)
 2000 FORMAT('.PROC,WIPCCL.')
C
C          LOOP OVER LINES IN AUDIT
      REWIND 1
      DO 100 L=1,5000
      READ(1,1000) ID, (PFN(I),I=1,4), CY, AC, DA
 1000 FORMAT(T6, A10,
     *        T18, 4A10,
     *        T60, A3,
     *        T67, A4,
     *        T125, A10)
      IF(EOF(1).NE.0) GO TO 101
C
C          DOES THIS LINE IDENTIFY A PERMFILE, AND IS THE
C          ACCOUNT NUMBER ELIGIBLE
      DO 40 IAC=1,NAC
      IF(AC.EQ.ACCOUNT(IAC)) GO TO 41
   40 CONTINUE
      GO TO 100
C
C          PURGE THE FILE.
   41 CALL CLEAR(A)
      A(1) = 'PURGE,X,'
      A(2) = PFN(1)
      A(3) = PFN(2)
      A(4) = PFN(3)
      A(5) = PFN(4)
      A(6) = ',ID='
      A(7) = ID
      A(8) = '.'
      CALL CLOSE(A)
      WRITE(2,2010) (A(I),I=1,12)
 2010 FORMAT(12A10)
C
```

```
C          RETURN FILE
      WRITE(2,2020)
 2020 FORMAT('RETURN,X.')
C
C          ERROR TRAP
      WRITE(2,2060)
 2060 FORMAT('EXIT,U.')
C
C          END OF AUDIT LINE LOOP
  100 CONTINUE
  101 CONTINUE
C
C          TERMINATE PROCEDURE
      WRITE(2,2070)
 2070 FORMAT('REVERT.')
C
      STOP
      END
      SUBROUTINE CLEAR(A)
C
C PURPOSE -- REPLACE A WITH BLANKS
C
      DIMENSION A(12)
C
      DO 100 I=1,12
      A(I) = ' '
  100 CONTINUE
C
      RETURN
      END
      SUBROUTINE CLOSE(A)
C
C PURPOSE -- REMOVE BLANKS AND SLASHES FROM ARRAY A
C
      DIMENSION A(12)
      DIMENSION B(120)
```

```
C
C          UNPACK
      DECODE(120,3000,A) (B(I),I=1,120)
 3000 FORMAT(120A1)
C
C          REMOVE CHARACTERS
      I2 = 0
      DO 100 I1=1,120
      IF(B(I1).EQ.' '.OR.
     *   B(I1).EQ.'/') GO TO 100
      I2 = I2 + 1
      B(I2) = B(I1)
      IF(B(I1).EQ.',') GO TO 110
  100 CONTINUE
C
C          PACK
  110 ENCODE(120,3000,A) (B(I),I=1,I2)
C
      RETURN
      END
.DATA,CAT.
      PROGRAM CAT(CATAUD,CATCCL,
     *   TAPE1=CATAUD,TAPE2=CATCCL,
     *   OUTPUT,TAPE6=OUTPUT)
C
C PURPOSE -- GENERATE A CCL PROCEDURE TO CATALOG EVERYONES
C            PERMILES ON MFA UNDER CERTAIN ACCOUNT NUMBERS.
C            FILES ARE CATALOGED UNDER ID=WOHL IN PREPARATION
C            FOR BACKUP USING DUMPF.
C
      IMPLICIT INTEGER(A-Z)
C
      DIMENSION A(12)
      DIMENSION ACCOUNT(5)
      DIMENSION PFN(4)
      DIMENSION CHAR(40)
```

```
C
C          SET ACCOUNT NUMBERS TO BACK UP
      DATA NAC /1/
      DATA ACCOUNT /'1349', 4*' '/
C
C          SET NEW ID
      DATA NEWID /'WMHL'/
C
C          WRITE PROCEDURE HEADER
      REWIND 2
      WRITE(2,2000)
 2000 FORMAT('.PROC,CATCCL,VSN.')
C
C          LOOP OVER LINES IN AUDIT
      REWIND 1
      DO 100 L=1,5000
      READ(1,1000) ID, (PFN(I),I=1,4), CY, AC, DA
 1000 FORMAT(T6, A10,
     *       T18, 4A10,
     *       T60, A3,
     *       T67, A4,
     *       T125, A10)
      IF(EOF(1).NE.0) GO TO 101
C
C          DOES THIS LINE IDENTIFY A PERMFILE, AND IS THE
C          ACCOUNT NUMBER ELIGIBLE
      DO 40 IAC=1,NAC
      IF(AC.EQ.ACCOUNT(IAC)) GO TO 41
   40 CONTINUE
      GO TO 100
C
C          DO NOT BACK UP FILES ALREADY ARCHIVED BY SYSTEM
   41 READ(1,1010) SYSARC
 1010 FORMAT(T81, A1)
      IF(SYSARC.EQ.'A') GO TO 100
C
```

```
217)      C          THIS IS A PERMFILE TO BE BACKED UP.
218)      C          ATTACH THE FILE.
219)            CALL CLEAR(A)
220)            A(1) = 'ATTACH,X,'
221)            A(2) = PFN(1)
222)            A(3) = PFN(2)
223)            A(4) = PFN(3)
224)            A(5) = PFN(4)
225)            A(6) = ',ID='
226)            A(7) = ID
227)            A(8) = ',CY='
228)            A(9) = CY
229)            A(10) = ',MR=1.'
230)            CALL CLOSE(A)
231)            WRITE(2,2010) (A(I),I=1,12)
232)       2010 FORMAT(12A10)
233)      C
234)      C          COPY FILE
235)            WRITE(2,2020)
236)       2020 FORMAT('REQUEST,Y,*PF.')
237)            WRITE(2,2030)
238)       2030 FORMAT('COPY,X,Y.')
239)      C
240)      C          CATALOG
241)            CALL CLEAR(A)
242)            A(1) = 'CATALOG,Y,'
243)            A(2) = ID
244)            A(3) = PFN(1)
245)            A(4) = PFN(2)
246)            A(5) = PFN(3)
247)            A(6) = PFN(4)
248)            CALL CLOSE(A)
249)            A(5) = DA
250)            A(6) = 'C'
251)            A(7) = CY
252)            CALL CLOSE(A)
```

```
      A(8) = ',ID='
      A(9) = NEWID
      A(10) = ',RP=1.'
      CALL CLOSE(A)
      WRITE(2,2040) (A(I),I=1,12)
 2040 FORMAT(12A10)
C
C          RETURN FILES
      WRITE(2,2050)
 2050 FORMAT('RETURN,X,Y.')
C
C          ERROR TRAP
      WRITE(2,2060)
 2060 FORMAT('EXIT,U.')
C
C          END OF AUDIT LINE LOOP
  100 CONTINUE
  101 CONTINUE
C
C          TERMINATE PROCEDURE
      WRITE(2,2070)
 2070 FORMAT('REVERT.')
C
      STOP
      END
      SUBROUTINE CLEAR(A)
C
C PURPOSE -- REPLACE A WITH BLANKS
C
      DIMENSION A(12)
C
      DO 100 I=1,12
      A(I) = ' '
  100 CONTINUE
C
      RETURN
```

```
      END
      SUBROUTINE CLOSE(A)
C
C PURPOSE -- REMOVE BLANKS AND SLASHES FROM ARRAY A
C
      DIMENSION A(12)
      DIMENSION B(120)
C
C          UNPACK
      DECODE(120,3000,A) (B(I),I=1,120)
 3000 FORMAT(120A1)
C
C          REMOVE CHARACTERS
      I2 = 0
      DO 100 I1=1,120
      IF(B(I1).EQ.' '.OR.
     *   B(I1).EQ.'/') GO TO 100
      I2 = I2 + 1
      B(I2) = B(I1)
      IF(B(I1).EQ.'.') GO TO 110
  100 CONTINUE
C
C          PACK
  110 ENCODE(120,3000,A) (B(I),I=1,I2)
C
      RETURN
      END
.DATA,LIST
      PROGRAM LIST(LISAUD,TAPE1=LISAUD,
     *   OUTPUT,TAPE6=OUTPUT)
C
C PURPOSE -- CREATE AN ALPHABETICAL LISTING OF BACKED UP FILES
C
      IMPLICIT INTEGER(A-Z)
C
      LEVEL 2, S
```

```
      COMMON /S/ S(4,1000), COUNT
C
      DIMENSION A(12)
      DIMENSION ACCOUNT(5)
      DIMENSION PFN(4)
      DIMENSION CHAR(40)
C
C          SET ACCOUNT NUMBERS TO BACK UP
      DATA NAC /1/
      DATA ACCOUNT /'1349', 4*' '/
C
C          LOOP OVER LINES IN AUDIT
      REWIND 1
      DO 100 L=1,5000
      READ(1,1000) ID, (PFN(I),I=1,4), CY, AC, DA
 1000 FORMAT(T6, A10,
     *       T18, 4A10,
     *       T60, A3,
     *       T67, A4,
     *       T125, A10)
      IF(EOF(1).NE.0) GO TO 101
C
C          DOES THIS LINE IDENTIFY A PERMFILE, AND IS THE
C          ACCOUNT NUMBER ELIGIBLE
      DO 40 IAC=1,NAC
      IF(AC.EQ.ACCOUNT(IAC)) GO TO 41
   40 CONTINUE
      GO TO 100
   41 CONTINUE
C
C          SAVE THE PERMFILE NAME
      CALL SAVE(PFN)
C
C          END OF AUDIT LINE LOOP
  100 CONTINUE
  101 CONTINUE
```

```
C
C          PRINT THE PERMFILE NAMES IN ALPHABETICAL ORDER
      WRITE(6,6000)
 6000 FORMAT('1FILES SAVED' ///)
      DO 200 F=1,COUNT
      DECODE(40,3000,S(1,F)) (CHAR(I),I=1,40)
 3000 FORMAT(40A1)
      DO 190 I=1,40
      IF(CHAR(I).EQ.':') CHAR(I) = ' '
  190 CONTINUE
      WRITE(6,6010) F, (CHAR(I),I=1,40)
 6010 FORMAT(' ', I4, 5X, 40A1)
  200 CONTINUE
C
      STOP
      END
      SUBROUTINE SAVE(V)
C
C PURPOSE -- SAVE AND SORT THE NEW PERMFILE NAME
C
      IMPLICIT INTEGER(A-Z)
C
      LEVEL 2, S
      COMMON /S/ S(4,1000), COUNT
C
      DIMENSION V(4)
      DIMENSION CHAR(40)
      DIMENSION Q(4)
C
C          INITIALIZE FILE COUNTER
      DATA COUNT /1/
      DATA S /'*END', 3*' '/
C
C          TRANSLATE BLANKS TO ZEROES
      DECODE(40,3000,V) (CHAR(I),I=1,40)
 3000 FORMAT(40A1)
```

```
      DO 20 I=1,40
      IF(CHAR(I).EQ.' ') CHAR(I) = 0
   20 CONTINUE
      ENCODE(40,3000,Q) (CHAR(I),I=1,40)
C
C          FIND CORRECT PLACE IN FILE
   50 DO 100 PLACE=1,COUNT
      DO 60 WORD=1,4
      IF(S(WORD,PLACE).LT.Q(WORD)) GO TO 100
      IF(S(WORD,PLACE).GT.Q(WORD)) GO TO 101
   60 CONTINUE
  100 CONTINUE
  101 CONTINUE
C
C          SHIFT ALL NAMES WHICH WILL COME AFTER THIS ONE
      NSHIFT = COUNT - PLACE + 1
      DO 150 I=1,NSHIFT
      IOLD = COUNT - I + 1
      INEW = IOLD + 1
      S(1,INEW) = S(1,IOLD)
      S(2,INEW) = S(2,IOLD)
      S(3,INEW) = S(3,IOLD)
      S(4,INEW) = S(4,IOLD)
  150 CONTINUE
C
C          ENTER NEW NAME
  200 S(1,PLACE) = Q(1)
      S(2,PLACE) = Q(2)
      S(3,PLACE) = Q(3)
      S(4,PLACE) = Q(4)
C
C          ADVANCE COUNTER
      COUNT = COUNT + 1
C
  900 RETURN
      END
```

Appendix B. ARC Output

FILES SAVED

1 KENANAAAAABBBBBCCCCCDDDDDEEEEE062982C1
2 KENANKENANAEPWRUO2RECYCLE063082C1
3 KENANKENANK7HAZARDINDEX063082C1
4 KENANKENANLMICRP30NUCLIDEDATA070182C1
5 KENANKENANW1PWRMOXFUEL063082C1
6 KENANKENANW1PWRSPENTFUEL063082C1
7 KENANNUCDAT5061682C1
8 KENANP5070782C1
9 KENANP8070882C1
10 KENANREP3060882C1
11 KENANR4060382C1
12 KENANT13E062282C1
13 KENANT13060182C1
14 KNAPPPERMA052782C1
15 KNAPPSMPL5052782C1
16 LLLHANFORD3LAYERSBOB031982C6
17 LLLSTLINERHOX122881C2
18 LLLSWIFTRHOSTLINE121181C1
19 LLLSWIFTRHOX050682C5
20 LLLSWIFTRHO121481C10
21 LLLSWIFTRIVAIF122381C1
22 LLLSWIFTRIVIAF122381C1
23 ORNSTEINAIFCALCULATIONS060382C2
24 ORNSTEINANALYT1A080481C10
25 ORNSTEINANALYT1B081981C1
26 ORNSTEINANALYT1091181C9
27 ORNSTEINIMAGE120781C18
28 ORNSTEINNWFTDVM1031582C3
29 ORNSTEINNWFTDVM2031682C1
30 ORNSTEINNWFTDVM3031782C2
31 ORNSTEINPARADOXSALTTRANSIENT062582C10
32 ORNSTEINPARADOXSALT061182C7
33 ORNSTEINPARADOXUSGS062482C5
34 ORNSTEINPARADOXUSGS062882C10
35 ORNSTEINPARADOX051482C10
36 ORNSTEINPARADOX052682C12
37 ORNSTEINSINP3120781C2
38 ORNSTEINSWIFTD1ORNSTEIN072481C6
39 ORNSTEINSWIFTUPDATES072281C1
40 QUINNDKRHOXPNLSE062482C1
41 QUINNDKRHOXPNLSE062882C2
42 QUINNDKRHOXPNLSE062882C3
43 QUINNDKRHOXPNLSE062982C4
44 QUINNDKRHOX051882C1
45 QUINNDKRHOX062882C2
46 QUINNDKRHOX2062982C1
47 QUINNDNET1A042781C1
48 QUINNFACTORFIRSTRUN011882C1
49 QUINNFIRSTFACTORRUN021082C6
50 QUINNHANFORDPNLBOUND021082C5
51 QUINNHANFORDPNLDK021882C1
52 QUINNHANFORDPNL021882C5
53 QUINNHANFORD3LAYERLESSAIF050582C4
54 QUINNRHOHANFORDFAULT050782C4
55 QUINNSTLINEPNL030182C3
56 QUINNSTLINEPNL051882C4
57 QUINNSWIFTRHOXFILLIN031282C9
58 SASACTUNCLE051682C1
59 SASARC062982C38