NUREG/CR–5667
EGG–2634

INEL Personal Computer Version of MACCS 1.5

Prepared by
K. R. Jones, C. A. Dobbe, D. L. Knudson

Idaho National Engineering Laboratory
EG&G Idaho, Inc.

Prepared for
U.S. Nuclear Regulatory Commission

NUREG/CR--5667

TI91 009217

INEL Personal Computer Version of MACCS 1.5

Manuscript Completed: December 1990
Date Published: March 1991

Prepared by
K. R. Jones, C. A. Dobbe, D. L. Knudson

Idaho National Engineering Laboratory
Managed by the U.S. Department of Energy

EG&G Idaho, Inc.
Idaho Falls, ID 83415

Prepared for
Division of Systems Research
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555
NRC FIN A6896
Under DOE Contract No. DE-AC07-76ID01570

ABSTRACT

The MELCOR Accident Consequence Code System, Version 1.5, (MACCS 1.5) calculates potential consequences resulting from atmospheric releases of radioactive materials. Sandia National Laboratories developed the code for the U.S. Nuclear Regulatory Commission on a VAX/VMS mini–computer. This report documents the Idaho National Engineering Laboratory conversion of MACCS 1.5 for compilation and execution on an 80386–based IBM or IBM–compatible personal computer (PC). The resulting PC version of the code is available through the National Energy Software Center, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL, 60439.

FIN No. A6896—Quality Assurance (QA) and Verification
of the MACCS Code, Version 1.5.

CONTENTS

FIGURES

TABLES

INEL PERSONAL COMPUTER VERSION OF MACCS 1.5

EXECUTIVE SUMMARY

The MELCOR Accident Consequence Code System, Version 1.5, (MACCS 1.5) calculates potential health and economic consequences resulting from atmospheric releases of radioactive materials. The code is a single FORTRAN–77 program with three basic modules–ATMOS, EARLY, and CHRONC. Sandia National Laboratories (SNL) developed the code for the U.S. Nuclear Regulatory Commission on a VAX/VMS mini–computer.

This report documents the Idaho National Engineering Laboratory (INEL) conversion of MACCS 1.5 for compilation and execution on an 80386–based IBM or IBM–compatible personal computer (PC). The coding modifications primarily consist of changes in system–dependent subroutine calls and the addition of a file handling processor. No modifications were made to the consequence calculation portions of MACCS 1.5.

A line–by–line check of all coding modifications was performed to ensure that the modifications were correct and to ensure that the modifications would not affect consequence calculations. In addition, a MACCS 1.5 sample problem was run using the INEL PC version of the code. The results were then compared to results generated by SNL using the VAX/VMS version. INEL PC version results were found to be identical to SNL VAX/VMS version results after accounting for round–off and truncation.

The PC version of the code is available through the National Energy Software Center, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL, 60439. This document contains a description of code transmittal content and instructions for loading and executing the INEL PC version of MACCS 1.5.

1. INTRODUCTION

The MELCOR Accident Consequence Code System, Version 1.5, (MACCS 1.5)[1] calculates potential health and economic consequences resulting from atmospheric releases of radioactive materials. The code consists of a single FORTRAN–77 program with three basic modules–ATMOS, EARLY, and CHRONC. Sandia National Laboratories (SNL) developed the code for the U.S. Nuclear Regulatory Commission (NRC) on a VAX/VMS minicomputer. This report documents Idaho National Engineering Laboratory (INEL) modifications required to compile and execute MACCS 1.5 on an 80386–based IBM or IBM–compatible personal computer (PC).

Coding modifications required to create the INEL PC version of MACCS 1.5 are outlined in Section 2. A complete discussion of the associated quality assurance program is also included. The resulting PC version of the code is available through the National Energy Software Center. Section 3 contains a description of the pertinent code transmittal content. PC system requirements and instructions for code execution are outlined in Section 4. References are provided in Section 5.

2. CODING MODIFICATIONS AND QUALITY ASSURANCE

All coding modifications required to create the INEL PC version of MACCS 1.5 (for the hardware and software configuration described in Section 4) are listed in Appendix A. The modifications primarily consist of changes in system–dependent subroutine calls and the addition of a file handling processor.

No modifications were made to the consequence calculation portions of MACCS 1.5. Therefore, calculational results of the code are not affected by the modifications. Two methods were used to ensure that consequence calculation portions of the code were not altered. First, a line–by–line check of the modifications was made to verify that all modifications were correctly coded. Second, sample problem results were generated with the INEL PC version and compared to results distributed with the SNL VAX/VMS version of MACCS 1.5.

Sample Problem A, as described in the MACCS 1.5 documentation (Reference 1), was used to perform the results comparison. All input files required to run the sample problem with the INEL PC version, the INEL PC version output file, and the sample problem output file generated on the SNL VAX/VMS mini–computer are provided in the PC code transmittal package. The code transmittal package also includes a listing of all differences that were determined by a line–by–line comparison of the two output files. (See Section 3, Code Transmittal Content.)

Only insignificant differences were identified through the line–by–line comparison of the output files. Those differences were caused by variations in round–off and truncation. In other words, the output file comparison verified that coding modifications did not corrupt calculational results.

Because the coding modifications were based on PC system dependencies and did not impact calculational results, the INEL PC and SNL VAX/VMS versions of MACCS 1.5 are identical for all practical purposes. On that basis, the quality assurance and verification of the SNL VAX/VMS version of MACCS 1.5[2] is also applicable to the INEL PC version.

3. CODE TRANSMITTAL CONTENT

The INEL PC version of MACCS 1.5 is available through the National Energy Software Center, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL, 60439. The transmittal is provided on three 3–1/2–in., high density diskettes. Code transmittal content is described below.

Diskette	File Name	File Description
1	READ.ME	An ASCII text file containing a description of code transmittal content, system requirements, and execution instructions.
1	INSTALL.BAT	A DOS batch file which will automatically load all transmitted files. An interactive option to execute Sample Problem A is provided.
1	MACCS.EXE	MACCS 1.5 as compiled for direct execution on an 80386–based IBM or IBM–compatible PC. This executable file was compiled with Version 3.01 of the Lahey F77L–EM/32 FORTRAN–77 compiler. As a result, it can be used without access to the A. I. Architects OS/386 DOS extender. Compiler options used are listed in Appendix A.
2	MACCS.FOR	FORTRAN–77 source code for the INEL PC version of MACCS 1.5.
3	SAMPLE.BAT	DOS batch file for optional execution of Sample Problem A.
3	SAMPLE.RUN	JCL/ATMOS input file for Sample Problem A.
3	IN2A10	EARLY input file for Sample Problem A.
3	IN3A10	CHRONC input file for Sample Problem A.
3	METSUR	Meteorological data file for Sample Problem A.
3	SURSI10	Site data file for Sample Problem A.
3	DOSDAT8	Dose conversion data file for MACCS 1.5.
3	LISTA11.OUT	SNL VAX/VMS Sample Problem A output file.
3	LISTAPC.OUT	INEL PC version Sample Problem A output file.
3	DIFF.OUT	Differences between SNL and INEL output files.

4. SYSTEM REQUIREMENTS AND CODE EXECUTION

4.1 Hardware

The minimum hardware configuration necessary to execute the INEL PC version of MACCS 1.5 includes:

- An Intel 80386–based IBM or IBM–compatible PC,
- An Intel 80387 math co–processor,
- A minimum of 3.3 megabytes of extended memory, and
- A minimum of 10 megabytes of free space on the hard disk.

4.2 Software

As discussed in Section 3, the code transmittal package includes a compiled executable file (MACCS.EXE) for the direct use of MACCS 1.5 on a PC. Assuming that the compiler options (listed in Appendix A) are acceptable, the minimum software necessary to execute the INEL PC version of MACCS 1.5 includes:

- MS–DOS or PC–DOS, Version 3.0 or higher.

If re–compiling is deemed necessary (i.e., compiler options for MACCS.EXE are unacceptable), the minimum software requirements must expand to include:

- The Lahey F77L–EM/32 FORTRAN–77 compiler, Version 1.01 or higher, and
- The A. I. Architects OS/386 DOS extender.

It should be noted that the Lahey F77L–EM/32 compiler is only required for compiling MACCS 1.5. For compiler versions below 3.01, access to A. I. Architects OS/386 DOS extender is required during execution. However, for compiler versions 3.01 (and above) as used in developing the INEL PC version of MACCS 1.5, execution access is unnecessary since the DOS extender is automatically bound to the executable file.

4.3 Execution

Before executing the INEL PC version of MACCS 1.5, the program must be loaded onto a PC with the required hardware and software configuration. The code transmittal package includes a file (INSTALL.BAT) to automate the loading process (see Section 3). To load all transmitted files, simply insert Diskette 1 into disk drive 'a'. From disk drive 'c', type 'a:\install' and follow the instructions on the screen. After all files have been loaded, the option to execute Sample Problem A will be offered.

To execute any other problem of interest, the user must develop appropriate input files according to the MACCS User's Guide (Reference 1). A job control language (JCL) structure is then used to provide the file handling required during execution of MACCS 1.5 on a PC. An example of the JCL developed for the INEL PC version of MACCS 1.5 is shown in Figure 1. Note that the ATMOS input file simply follows the JCL as indicated. The listed JCL can be duplicated or the JCL can be stripped from the beginning of the file 'SAMPLE.RUN'.

JCL comment lines (like most comment lines in MACCS 1.5 input files) are preceded by an asterisk. The JCL is separated from the ATMOS input data by the line starting with a '/*'. JCL file specifications should not exceed 72 characters in length. If necessary, JCL file specifications may be continued onto the next line through the use of a comma as shown in the figure.

File specifications are designed to allow flexibility in code input/output. The user should be careful to provide the appropriate path names to the files necessary for the MACCS 1.5 problem of interest. Unnecessary file specification(s) must

```
*************************************************************************
*  MELCOR ACCIDENT CONSEQUENCE CODE SYSTEM (MACCS)          VER 1.5  *
*-----------------------------------------------------------------------*
*                                                                       *
*  CASE DESCRIPTION:  SAMPLE PROBLEM A FROM SNL, SURRY MODEL            *
*                                                                       *
*  UNIT   FILE DESCRIPTION                                              *
*  ----   ---------------                                               *
*   --    Input:                                                        *
*   25      EARLY User Input File                                       *
*   26      CHRONC User Input File                                      *
*   27      DOSE Factors Input Data File                                *
*   28      Meteorological Input Data File                              *
*   29      Site Input Data File                                        *
*   --    Output:                                                       *
*    6      STANDARD PRINTER OUTPUT                                     *
*                                                                       *
*  INPUT:  ATMOS User Input File (UNIT 24)                              *
*                                                                       *
*************************************************************************
* GOESINS:
FILE25='C:\MACCS\SAMPLE\in2a10',   STATUS='OLD',  ACTION='READ'
FILE26='C:\MACCS\SAMPLE\in3a10',   STATUS='OLD',  ACTION='READ'
FILE27='C:\MACCS\SAMPLE\dosdat8',  STATUS='OLD',  ACTION='READ'
FILE28='C:\MACCS\SAMPLE\metsur',   STATUS='OLD',  ACTION='READ'
FILE29='C:\MACCS\SAMPLE\sursi10',  STATUS='OLD',  ACTION='READ'
* GOESOUTS:
FILE06='C:\MACCS\SAMPLE\sample_a.out',  STATUS='UNKNOWN',
                                CARRIAGE CONTROL='FORTRAN'
/*
*
( append ATMOS input file here )
```

Figure 1. An example of the JCL developed for the INEL PC version of MACCS 1.5.

be deleted or 'commented out'. Complete discussions of the file specification options can be found in the Lahey F77L–EM/32 compiler documentation and in Appendix A.

At the DOS prompt, execution of the problem of interest is accomplished by typing 'c:\maccs\maccs *atmos.inp*' where '*atmos.inp*' is the name (or path name) of the file containing the JCL and appended ATMOS input data. A comparison of MACCS 1.5 processing times (in central processing unit seconds) for several machines is given in Table 1.

Table 1. A comparison of MACCS 1.5 processing times for several machines

Computer	CPU Time (s)
INEL Cray X–MP/24	286.1
SNL VAX/VMS	1392.3
IBM PS/2 Model 70 E61	10719.8
Apollo 3500	18197.0

5. REFERENCES

1. D. I. Chanin et al., *MELCOR Accident Consequence Code System (MACCS)*, NUREG/CR–4691, SAND86–1562, Volumes 1–3, February 1990.

2. C. A. Dobbe et al., *Quality Assurance and Verification of the MACCS Code, Version 1.5*, NUREG/CR–5376, EGG–2566, February 1990.

APPENDIX A

CODING MODIFICATIONS

APPENDIX A

CODING MODIFICATIONS

This appendix contains a list of the differences between the original SNL MACCS 1.5 code and the INEL PC version of the code. This appendix also contains a list of recommended compiler options.

A.1 DIFFERENCES

The key below is an explanation of the types of changes that were made to the original code (MACCS11.ORG) to obtain the new code (MACCS11.FOR).

- "<" in column 1 indicates that the line was removed from MACCS11.ORG.
- ">" in column 1 indicates that the line was added to MACCS11.ORG.

```
TO TRANSFORM maccs11.org INTO maccs11.for ...

*** CHANGE 618 IN maccs11.org TO [618,619] IN maccs11.for ***
<       DATA REVNAM /'TEST VERSION 1.5.11, 10/17/89, D. CHANIN'/

---------------
>CKRJ   DATA REVNAM /'TEST VERSION 1.5.11, 10/17/89, D. CHANIN'/
>       DATA REVNAM /'INEL PC VERSION 1.5.11, 10/26/89, K. JONES'/        *KRJ

*** CHANGE 737 IN maccs11.org TO [738,740] IN maccs11.for ***
<       MACHIN = 'VAX/VMS'

---------------
>CKRJ   MACHIN = 'VAX/VMS'
>       MACHIN = 'IBM-PC/LAHEY'                                           *KRJ
>       CALL IOFILES(24,6)                                                *KRJ

*** CHANGE 742 IN maccs11.org TO 745 IN maccs11.for ***
<       OPEN (6, STATUS = 'UNKNOWN')

---------------
>CKRJ   OPEN (6, STATUS = 'UNKNOWN')

*** CHANGE 750 IN maccs11.org TO 753 IN maccs11.for ***
<       SUBROUTINE MXXCPU (CPUTIM)

---------------
>CKRJ   SUBROUTINE MXXCPU (CPUTIM)
```

```
*** CHANGE [754,757] IN maccs11.org TO [757,760] IN maccs11.for ***
<       CHARACTER *80   MACHIN
<       COMMON /MACHIN/ MACHIN
<
<       DATA ICPUTC / 2 /

---------------
>CKRJ  CHARACTER *80   MACHIN
>CKRJ  COMMON /MACHIN/ MACHIN
>CKRJ
>CKRJ  DATA ICPUTC / 2 /

*** CHANGE 761 IN maccs11.org TO 764 IN maccs11.for ***
<       DATA FIRST / 0.0 /

---------------
>CKRJ  DATA FIRST / 0.0 /

*** CHANGE [765,770] IN maccs11.org TO [768,773] IN maccs11.for ***
<       IF (MACHIN .NE. 'VAX/VMS') CALL ABORT ('MXXCPU')
<
<       IF (FIRST .EQ. 0.) THEN
<         FIRST = 1.
<         CALL LIB$INIT_TIMER
<       ENDIF

---------------
>CKRJ  IF (MACHIN .NE. 'VAX/VMS') CALL ABORT ('MXXCPU')
>CKRJ
>CKRJ  IF (FIRST .EQ. 0.) THEN
>CKRJ    FIRST = 1.
>CKRJ    CALL LIB$INIT_TIMER
>CKRJ  ENDIF

*** CHANGE [774,779] IN maccs11.org TO [777,782] IN maccs11.for ***
<       CALL LIB$STAT_TIMER (ICPUTC,IECPUI)
<
<       CPUTIM = 0.01 * FLOAT (IECPUI)
<
<       RETURN
<       END

---------------
>CKRJ  CALL LIB$STAT_TIMER (ICPUTC,IECPUI)
>CKRJ
>CKRJ  CPUTIM = 0.01 * FLOAT (IECPUI)
>CKRJ
>CKRJ  RETURN
>CKRJ  END

*** CHANGE 790 IN maccs11.org TO [793,794] IN maccs11.for ***
<       IF (MACHIN .EQ. 'VAX/VMS' ) THEN
```

```
---------------
>CKRJ  IF (MACHIN .EQ. 'VAX/VMS' ) THEN
>      IF (MACHIN .EQ. 'IBM-PC/LAHEY'  ) THEN                          *KRJ

*** CHANGE 810 IN maccs11.org TO [814,818] IN maccs11.for ***
<      IF ( MACHIN .EQ. 'VAX/VMS' ) THEN

---------------
>      CHARACTER*3 MONTH(12)                                           *KRJ
>      DATA MONTH /'JAN','FEB','MAR','APR','MAY','JUN',                *KRJ
>     $            'JUL','AUG','SEP','OCT','NOV','DEC'/                *KRJ
>                                                                      *KRJ
>CKRJ  IF ( MACHIN .EQ. 'VAX/VMS' ) THEN

*** CHANGE [814,815] IN maccs11.org TO [822,833] IN maccs11.for ***
<        CALL DATE (CHAR9)
<      ENDIF

---------------
>CKRJ    CALL DATE (CHAR9)
>CKRJ  ENDIF
>                                                                      *KRJ
>      IF ( MACHIN .EQ. 'IBM-PC/LAHEY' ) THEN                          *KRJ
>C                                                                     *KRJ
>C  FORM OF THE TEXT STRING RETURNED IN TODAY IS '11-01-88'            *KRJ
>C  FORM OF THE TEXT STRING RETURNED IN CHAR9 IS '11-JAN-88'           *KRJ
>C                                                                     *KRJ
>        CALL DATE (TODAY)                                             *KRJ
>        READ(TODAY,'(I2)') IMONTH                                     *KRJ
>        CHAR9 = TODAY(4:5)//'-'//MONTH(IMONTH)//'-'//TODAY(7:8)       *KRJ
>      ENDIF                                                           *KRJ

*** CHANGE 895 IN maccs11.org TO 913 IN maccs11.for ***
<      OPEN (24, STATUS='UNKNOWN')

---------------
>CKRJ  OPEN (24, STATUS='UNKNOWN')

*** CHANGE 983 IN maccs11.org TO 1001 IN maccs11.for ***
<        OPEN (25, STATUS='UNKNOWN')

---------------
>CKRJ    OPEN (25, STATUS='UNKNOWN')

*** CHANGE 1046 IN maccs11.org TO 1064 IN maccs11.for ***
<        OPEN (26, STATUS='UNKNOWN')

---------------
>CKRJ    OPEN (26, STATUS='UNKNOWN')

*** APPEND AFTER 1124 IN maccs11.org ***
>CKRJ If the ATMOS file is being read, move the file pointer to after  *KRJ
>CKRJ the JCL information.                                              *KRJ
>         IF(IUNIT.EQ.24) THEN                                         *KRJ
> 97          READ(IUNIT,1001) CARD(1)                                *KRJ
```

```
>             IF(CARD(1)(1:2).NE.'/*') GOTO 97                          *KRJ
>             CARD(1) = ' '                                             *KRJ
>          ENDIF                                                        *KRJ

*** APPEND AFTER 1164 IN maccs11.org ***
>CKRJ If the ATMOS file is being read, move the file pointer to after    *KRJ
>CKRJ the JCL information.                                               *KRJ
>          IF(IUNIT.EQ.24) THEN                                         *KRJ
> 99          READ(IUNIT,1001) CARD(1)                                 *KRJ
>             IF(CARD(1)(1:2).NE.'/*') GOTO 99                          *KRJ
>             CARD(1) = ' '                                             *KRJ
>          ENDIF                                                        *KRJ

*** CHANGE 4822 IN maccs11.org TO 4854 IN maccs11.for ***
<         OPEN (28, STATUS='UNKNOWN')

---------------
>CKRJ     OPEN (28, STATUS='UNKNOWN')

*** CHANGE 6270 IN maccs11.org TO 6302 IN maccs11.for ***
<       OPEN (27, STATUS='UNKNOWN')

---------------
>CKRJ   OPEN (27, STATUS='UNKNOWN')

*** CHANGE 6281 IN maccs11.org TO 6313 IN maccs11.for ***
<       OPEN (29, STATUS='UNKNOWN')

---------------
>CKRJ   OPEN (29, STATUS='UNKNOWN')
```

In addition to the changes listed above, subroutines IOFILES and MXXCPU, function JULDAY, and block data routine ONE were added to the program source code. The subroutine IOFILES was linked to MACCS 1.5.11 through a call from subroutine MXXECT to establish file handles based on user input from JCL type statements placed before the input file. This allows the user to specify file names (including the DOS path), file status, file access, file form, and carriage control file attributes for each FORTRAN unit. This subroutine also prints a banner page and file handling summary on the output, which contains the date and start time to uniquely identify the run. A "/*" card terminates the file specification list, and program input may follow. The subroutine requires two arguments that correspond to the primary input and output FORTRAN unit numbers for the program. The input file name containing the file specifications is read from the command line. The format for specification of file handles is as follows:

```
*************************************************************************
*  Comments can be entered by placing a '*' in column 1 and will be    *
*  echoed to the output file.                                           *
*                                                                       *
*  An example file specification format is shown below.  Only the UNIT *
*  number and file name are required input.  Columns 73 to 80 are       *
*  reserved for comments, a line ending in a comma is continued on      *
*  the following line.                                                  *
*************************************************************************
*
FILEnn='D:\path\filename',  STATUS='status' , ACCESS='access',
       FORM='form', CARRIAGE CONTROL='carriage'
*
/* end of JCL card
```

ATMOS Module input is as follows:

nn is the FORTRAN Unit number to be opened. Must be preceded by "FILE" in columns 1 to 4. Possible range of values is 01 to 99.

D:\path\filename is the DOS Drive, Path and Filename. DOS devices CON,LPT1,LPT2,LPT3,PRN,COM1,COM2, or NUL may also be used.

status is the open status of the file. Valid options include: 'NEW', 'OLD', 'UNKNOWN' or 'SCRATCH' corresponding to the FORTRAN-77 standard. The default value is 'UNKNOWN'.

access is the access method for the file. Valid options are 'SEQUENTIAL' and 'DIRECT' corresponding to the FORTRAN-77 standard. The default value is 'SEQUENTIAL'.

form indicates the file format and has two possible options: 'FORMATTED' and 'UNFORMATTED' (corresponding to the FORTRAN-77 standard).

carriage indicates whether the first character from each sequential formatted output record will be used as FORTRAN printer carriage control characters. Possible options are 'FORTRAN' and 'LIST'. The default values are 'LIST' for disk files and 'FORTRAN' for DOS devices.

***** indicates the end of the file processing input. The input file for the ATMOS module (Unit 24) is placed directly after this card.

Other coding added to MACCS 1.5.11 includes a replacement for the system-dependent subroutine, MXXCPU, which returns the current CPU time in seconds and a block data routine to zero out common-block MULREL. The block data routine, ONE, is necessary to prevent accessing uninitialized memory in subroutine PUTSTM during the comparison test, which is used to determine whether the input values represent a change in the source term data. The following is a listing of these additional routines that were appended to the end of file maccs11.for.

```
      SUBROUTINE IOFILES(IUNIT,IOUT)
C*************************************************************************
C IOFILES.FOR:  JCL TYPE FILE HANDLE PROCESSOR
C
C VERSION:      2.0 - LAHEY EM32
C
C DEVELOPED BY: K. R. JONES
C
C ARGUMENTS:    IUNIT - STANDARD INPUT UNIT NUMBER (FILENAME OBTAINED
C                       FROM COMMAND LINE OR PROMPT)
C               IOUT  - STANDARD OUTPUT UNIT NUMBER
C
C PROGRAMMING NOTES:
C
C 1. MODIFY DEFAULT FILE NAME, IOFILE, AND FORMAT STATEMENTS 500-503
C    AS REQUIRED TO ADAPT TO THE SPECIFIC PROGRAM.
C 2. IF IUNIT IS ALSO USED FOR PROGRAM INPUT, ENSURE THAT ANY
C    "REWIND(IUNIT)" STATEMENTS ARE REMOVED FROM THE SOURCE TO
C    PREVENT REWINDING INTO THE JCL.
C 3. THIS SOURCE CONTAINS NON-F77 STANDARD SYSTEM AND STRING MANIPULATION
C    CALLS FROM LAHEY FORTRAN.
C 4. COMMENT OUT ACTION SECTION FOR COMPATIBILITY WITH EARLIER LAHEY
C    COMPILERS
C
C NOTES ON JCL STRUCTURE:
C
C  Comments can be entered by placing a '*' in column 1 and will be
C  echoed to the output file.
C  An example file specification format is shown below.  Only the UNIT
C  number and file name are required input.  Columns 73 to 80 are
C  reserved for comments, a line ending in a comma is continued on
C  the following line.
C
C FILEnn='D:\path\filename',  STATUS='status' , ACCESS='access',
C         FORM='form', CARRIAGE CONTROL='carriage', ACTION='action'
C *
C /* end of JCL card
C program input may follow ...
C
C
```

```
C Where:
C   'nn'  is the FORTRAN Unit number to be opened.
C
C   'D:\path\filename' is the DOS Drive, Path and Filename or a
C         DOS device (CON,LPT1,LPT2,LPT3,PRN,COMn,NUL)
C
C   'status' is the open status of the file.
C
C   'access' is the access method for the file.
C
C   'form' indicates the file format.
C
C   'carriage' indicates whether FORTRAN printer carriage control
C              characters are used.
C
C   'action' specifies the permission level for the file.
C
C*************************************************************************
C
      CHARACTER*72 FILENM,IOFILE,CMDLINE,MESSAG
      CHARACTER*80  LINE
      CHARACTER*432 L1
      CHARACTER*14  STATUS
      CHARACTER*14  FORM,ACCESS
      CHARACTER*14  CARRIAGE
      CHARACTER*8   POSITION
      CHARACTER*22  ACTION
      CHARACTER*1   QUOTE
      CHARACTER*8   TODAY
      CHARACTER*11  CTIME
      LOGICAL LEXIST
      LOGICAL LSCAN
      INTEGER UNIT
      DATA QUOTE   / 1H' /
      IOFILE='MACCS.INP'
C
C  OPEN FILE HANDLE FILE USING COMMAND LINE INPUT OR PROMPT
C
      CALL GETCL(CMDLINE)
      IF(CHARNB(CMDLINE).EQ.' ') THEN
        CMDLINE=IOFILE
      ENDIF
   20 IOFILE=CHARNB(CMDLINE)
      INQUIRE(FILE=IOFILE,EXIST=LEXIST)
      IF(.NOT. LEXIST) THEN
        WRITE(*,300) IOFILE
        READ*,CMDLINE
        IF(CHARNB(CMDLINE) .EQ. 'Q'  .OR.
     *     CHARNB(CMDLINE) .EQ. 'q') GOTO 3
        GOTO 20
      ENDIF
      OPEN(IUNIT,FILE=IOFILE  ,IOSTAT=MSGNO,ERR=40,
     *     STATUS='OLD',FORM='FORMATTED')
C
```

```
C  GET CURRENT DATE AND TIME
C
      CALL DATE(TODAY)
      CALL TIME(CTIME)
C
C  READ INPUT LINE
C
      LSCAN =.TRUE.
   1  READ (IUNIT,100,END=2) LINE
C
      IF(LSCAN) THEN
        IF(LINE(1:1) .EQ. '*')  GOTO 1
        IF(LINE(1:2) .EQ. '/*') GOTO 44
      ELSE
        WRITE(IOUT,102) LINE
        IF(LINE(1:1) .EQ. '*')  THEN
          GOTO 1
        ENDIF
        IF(LINE(1:2) .EQ. '/*') THEN
          WRITE(IOUT,103)
          GOTO 2
        ENDIF
      ENDIF
      IF(LINE(1:4) .EQ. 'FILE') THEN
         READ(LINE,'(4X,I2)') UNIT
         IF(UNIT.LT.0 .OR. UNIT.GT.99) GOTO 41
         IF(UNIT.NE.IOUT .AND. LSCAN) THEN
           GOTO 1
         ENDIF
      ELSE
         GOTO 1
      ENDIF
      L1(1:72)=LINE(1:72)
      M=73
C ------CHECK FOR CONTINUATIONS-------------
    5 JLAST=6+INDEX(LINE(7:72),',')
   10 J=JLAST+INDEX(LINE(JLAST+1:72),',')
      IF(J.GT.JLAST) THEN
         JLAST=J
         GOTO 10
      ENDIF
      IF(CHARNB(LINE(JLAST+1:72)) .EQ. ' ') THEN
         READ (IUNIT,100,END=2) LINE
         IF(.NOT.LSCAN) WRITE(IOUT,102) LINE
         IF(LINE(1:2) .EQ. '/*') THEN
            WRITE(*,101)
            GOTO 3
         ELSE
            L1(M:M+72)=LINE(1:72)
            M=M+73
         ENDIF
         GOTO 5
      ENDIF
C ----- FILE NAME -----
```

```
      IFRST=INDEX(L1(1:432),QUOTE)
      ILAST=IFRST+INDEX(L1(IFRST+1:432),QUOTE)
      FILENM=L1(IFRST+1:ILAST-1)
      IF(CHARNB(FILENM).EQ.'NULLFILE') THEN
         STATUS='SCRATCH'
      ENDIF
C ----- FILE STATUS ---
      III=INDEX(L1,'STATUS')
      IF(III .GT. 0) THEN
         IFRST=III  +INDEX(L1(III  +1:432),QUOTE)
         ILAST=IFRST+INDEX(L1(IFRST+1:432),QUOTE)
         STATUS=L1(IFRST+1:ILAST-1)
      ELSE
         IF(CHARNB(FILENM).EQ.'NULLFILE') THEN
            STATUS='SCRATCH'
         ELSE
            STATUS='UNKNOWN'
         ENDIF
      ENDIF
C ----- FILE FORM -----
      III=INDEX(L1,'FORM')
      IF(III .GT. 0) THEN
         IFRST=III  +INDEX(L1(III  +1:432),QUOTE)
         ILAST=IFRST+INDEX(L1(IFRST+1:432),QUOTE)
         FORM=L1(IFRST+1:ILAST-1)
      ELSE
         FORM='FORMATTED'
      ENDIF
C ----- FILE ACCESS ---
      III=INDEX(L1,'ACCESS')
      IF(III .GT. 0) THEN
         IFRST=III  +INDEX(L1(III  +1:432),QUOTE)
         ILAST=IFRST+INDEX(L1(IFRST+1:432),QUOTE)
         ACCESS=L1(IFRST+1:ILAST-1)
      ELSE
         ACCESS='SEQUENTIAL'
      ENDIF
C ----- FILE CARRIAGE---
      III=INDEX(L1,'CARRIAGE CONTROL')
      IF(III .GT. 0) THEN
         IFRST=III  +INDEX(L1(III  +1:432),QUOTE)
         ILAST=IFRST+INDEX(L1(IFRST+1:432),QUOTE)
         CARRIAGE=L1(IFRST+1:ILAST-1)
      ELSE
         IF((ACCESS.EQ.'SEQUENTIAL').AND.(FORM.EQ.'FORMATTED')) THEN
            CARRIAGE='LIST'
         ELSE
            CARRIAGE='NONE'
         ENDIF
      ENDIF
C ----- FILE POSITION---
      III=INDEX(L1,'POSITION')
      IF(III .GT. 0) THEN
         IFRST=III  +INDEX(L1(III  +1:432),QUOTE)
```

```
         ILAST=IFRST+INDEX(L1(IFRST+1:432),QUOTE)
         POSITION=L1(IFRST+1:ILAST-1)
      ELSE
         IF(ACCESS.EQ.'SEQUENTIAL') THEN
            POSITION='ASIS'
         ELSE
            POSITION='NONE'
         ENDIF
      ENDIF
C ----- FILE ACTION ----
      III=INDEX(L1,'ACTION')
      IF(III .GT. 0) THEN
         IFRST=III  +INDEX(L1(III  +1:432),QUOTE)
         ILAST=IFRST+INDEX(L1(IFRST+1:432),QUOTE)
         ACTION=L1(IFRST+1:ILAST-1)
      ELSE
         IF(ACCESS.EQ.'SEQUENTIAL') THEN
            ACTION='READ/WRITE'
         ELSE
            ACTION='NONE'
         ENDIF
      ENDIF
C ---- OPEN FILE ----
      IF(LSCAN .OR. (.NOT.LSCAN .AND. IUNIT.NE.IOUT)) THEN
        IF(CHARNB(FILENM) .EQ. 'NULLFILE' .OR.
     *     CHARNB(STATUS) .EQ. 'SCRATCH') THEN
          IF(POSITION.NE.'NONE') THEN
            OPEN(UNIT,STATUS=STATUS,FORM=FORM,ACCESS=ACCESS,
     *         ERR=40,IOSTAT=MSGNO,
     *         POSITION=POSITION,ACTION=ACTION)
          ELSE
            OPEN(UNIT,STATUS=STATUS,FORM=FORM,ACCESS=ACCESS,
     *         ERR=40,IOSTAT=MSGNO,
     *         ACTION=ACTION)
          ENDIF
        ELSE
          IF(CHARNB(CARRIAGE) .NE. 'NONE') THEN
             IF(POSITION.NE.'NONE') THEN
               OPEN(UNIT,FILENM,STATUS=STATUS,FORM=FORM,ACCESS=ACCESS,
     *            CARRIAGE CONTROL=CARRIAGE,ERR=40,IOSTAT=MSGNO,
     *            POSITION=POSITION,ACTION=ACTION)
             ELSE
               OPEN(UNIT,FILENM,STATUS=STATUS,FORM=FORM,ACCESS=ACCESS,
     *            CARRIAGE CONTROL=CARRIAGE,ERR=40,IOSTAT=MSGNO,
     *            ACTION=ACTION)
             ENDIF
          ELSE
             IF(POSITION.NE.'NONE') THEN
               OPEN(UNIT,FILENM,STATUS=STATUS,FORM=FORM,ACCESS=ACCESS,
     *            ERR=40,IOSTAT=MSGNO,
     *            POSITION=POSITION,ACTION=ACTION)
             ELSE
               OPEN(UNIT,FILENM,STATUS=STATUS,FORM=FORM,ACCESS=ACCESS,
     *            ERR=40,IOSTAT=MSGNO,
```

```
     *            POSITION=POSITION,ACTION=ACTION)
               ENDIF
             ENDIF
           ENDIF
         ENDIF
         IF(LSCAN) THEN
           write(iout,501)
           write(iout,502)
           write(iout,503) TODAY,CTIME
           WRITE(*,500) TODAY,CTIME,IUNIT,IOFILE
           WRITE(IOUT,500) TODAY,CTIME,IUNIT,IOFILE
           REWIND(IUNIT)
           LSCAN=.FALSE.
         ELSE
           WRITE(*,210)UNIT,FILENM,STATUS,FORM,ACCESS,CARRIAGE
           WRITE(IOUT,210)UNIT,FILENM,STATUS,FORM,ACCESS,CARRIAGE
         ENDIF
         L1 = ' '
         LINE = ' '
         GOTO 1
C
    40 CONTINUE
       CALL IOSTAT_MSG(MSGNO,MESSAG)
       WRITE(*,200)UNIT,MSGNO,MESSAG,FILENM,STATUS,FORM,ACCESS,CARRIAGE
       WRITE(IOUT,200)UNIT,MSGNO,MESSAG,FILENM,STATUS,FORM,ACCESS,
      +               CARRIAGE
       GOTO 3
    41 WRITE(*,201) UNIT
       GOTO 3
    44 WRITE(*,203) IOUT
         WRITE(*,500) TODAY,CTIME,IUNIT,IOFILE
         REWIND(IUNIT)
         LSCAN=.FALSE.
       GOTO 1
C
     3 CONTINUE
       STOP
     2 CONTINUE
   100 FORMAT (A80)
   101 FORMAT (' EXPECTED CONTINUATION LINE NOT FOUND')
   102 FORMAT (' ',A80)
   103 FORMAT (' # END OF FILE PROCESSING',/,1H1)
   200 FORMAT(/' IOFILES: ERROR ON TRYING TO OPEN UNIT ',I2/
      *       12X,'ERROR MESSAGE NUMBER = ',I10,' ***'/
      *       2X,A72//
      *       2X,'FILE NAME: ',A60/
      *       2X,'STATUS   :',A15,
      *       2X,'FORM     :',A15/
      *       2X,'ACCESS   :',A15,
      *       2X,'CARRIAGE :',A15/)
   210 FORMAT(' # FORTRAN UNIT ',I2,' OPENED'/
      *       ' #    FILE NAME: ',A60/
      *       ' #    STATUS   :',A15,4X,'FORM     :',A15/
      *       ' #    ACCESS   :',A15,4X,'CARRIAGE :',A15)
```

```
201 FORMAT(/' IOFILES: ERROR ON TRYING TO OPEN UNIT NUMBER: ',I10)
202 FORMAT(/' IOFILES: ERROR TRYING TO INTERPRET FILE HANDLES')
203 FORMAT(/' IOFILES: OUTPUT FILE UNIT:',I2,' SPECIFICATION NOT FOUND
   *'/      ' STD OUTPUT WILL BE ROUTED TO CONSOLE')
300 FORMAT(/' THE INPUT FILE COULD NOT BE LOCATED.',
   *       /' FILE:',A72
   *       /' REENTER FILE SPECIFICATION OR "Q" TO QUIT.'
   *       /' >')
500 FORMAT('1MELCOR ACCIDENT CONSEQUENCE CODE SYSTEM (MACCS) ',
   +        'VERSION 1.5.11',
   +       /'    CURRENT DATE: ',A8,'  TIME: ',A11,
   +       /'    UNIT: ',I3,'  INPUT FILE: ',A30,/)
501 FORMAT(////,
   +20X,'MM      MM     AAAA      CCCCCCCCC   CCCCCCCCC   SSSSSSSSS'/
   +20X,'MMM    MMM    AAAAAA    CCCCCCCCCC  CCCCCCCCCC  SSSSSSSSSS'/
   +20X,'MMMM  MMMM   AA    AA   CC          CC          SS        '/
   +20X,'MM MMMM MM  AA      AA  CC          CC          SS        '/
   +20X,'MM  MM  MM  AAAAAAAAAA  CC          CC          SSSSSSSSSS'/
   +20X,'MM      MM  AAAAAAAAAA  CC          CC          SSSSSSSSSS'/
   +20X,'MM      MM  AA      AA  CC          CC                  SS'/
   +20X,'MM      MM  AA      AA  CC          CC                  SS'/
   +20X,'MM      MM  AA      AA  CCCCCCCCCC  CCCCCCCCCC  SSSSSSSSSS'/
   +20X,'MM      MM  AA      AA   CCCCCCCCC   CCCCCCCCC  SSSSSSSSS ')
502 FORMAT(//,
   +20X,'                  PPPPPPPPP    CCCCCCCCC                  '/
   +20X,'                  PPPPPPPPPP  CCCCCCCCCC                  '/
   +20X,'                  PP      PP  CC                          '/
   +20X,'                  PP      PP  CC                          '/
   +20X,'     \\\\\\\\\\   PPPPPPPPPP  CC          /////////       '/
   +20X,'     //////////   PPPPPPPPP   CC          \\\\\\\\\       '/
   +20X,'                  PP          CC                          '/
   +20X,'                  PP          CC                          '/
   +20X,'                  PP          CCCCCCCCCC                  '/
   +20X,'                  PP           CCCCCCCCC                  '/
   +20X,//,
   +20X,'                                                          ')
503 FORMAT(////,
   +15X,'VV      VV  EEEEEEEEEE  RRRRRRRRR           11          ',
   +'5555555555'/
   +15X,'VV      VV  EEEEEEEEEE  RRRRRRRRRR         111          ',
   +'5555555555'/
   +15X,'VV      VV  EE          RR      RR        1111          ',
   +'55        '/
   +15X,'VV      VV  EE          RR      RR          11          ',
   +'55        '/
   +15X,'VV      VV  EEEEEEEEEE  RRRRRRRRRR  \\\     11          ',
   +'555555555 '/
   +15X,'VV      VV  EEEEEEEEEE  RRRRRRRRR   \\\     11          ',
   +'5555555555'/
   +15X,' VV    VV   EE          RR   RR             11          ',
   +'        55'/
   +15X,'  VV  VV    EE          RR    RR            11          ',
   +'        55'/
   +15X,'   VVVV     EEEEEEEEEE  RR     RR        1111111111  OO  ',
```

```
     +'5555555555'/
     +15X,'    VV      EEEEEEEEEE  RR      RR       1111111111  00  ',
     +'555555555 '/
     +15X,//,
     +15X,'                                                          '/
     +/////32X,'RUN ON: ',A8,'  TIME: ',A11)
  600 FORMAT(/' ENTER FILE NAME OR "Q" TO QUIT.',
     *       /' >')
      RETURN
      END

      SUBROUTINE MXXCPU(SECONDS)
      SAVE
      CHARACTER*8   STARTDAY,TODAY
      DATA IFIRST /0/
      IF(IFIRST.EQ.0) THEN
         CALL TIMER(I100STRT)
         CALL DATE(STARTDAY)
         READ(STARTDAY,'(I2,1X,I2,1X,I2)') IM,ID,IY
         IY = IY + 1900
         JULSTART = JULDAY(IM,ID,IY)
         IFIRST = 1
         SECONDS = 0.0
      ELSE
         CALL TIMER(I100NOW)
         CALL DATE(TODAY)
         READ(TODAY,'(I2,1X,I2,1X,I2)') IM,ID,IY
         IY = IY + 1900
         JULNOW = JULDAY(IM,ID,IY)
         IFIRST = 1
         SECONDS = FLOAT(JULNOW-JULSTART)*86400.
     +             + FLOAT(I100NOW-I100STRT)/100.
      ENDIF
      RETURN
      END

      FUNCTION JULDAY(MM,ID,IYYY)
      PARAMETER (IGREG=15+31*(10+12*1582))
      IF (IYYY.EQ.0) PAUSE 'There is no Year Zero.'
      IF (IYYY.LT.0) IYYY=IYYY+1
      IF (MM.GT.2) THEN
        JY=IYYY
        JM=MM+1
      ELSE
        JY=IYYY-1
        JM=MM+13
      ENDIF
      JULDAY=INT(365.25*JY)+INT(30.6001*JM)+ID+1720995
      IF (ID+31*(MM+12*IYYY).GE.IGREG) THEN
        JA=INT(0.01*JY)
        JULDAY=JULDAY+2-JA+INT(0.25*JA)
      ENDIF
      RETURN
      END
```

```
      BLOCK DATA ONE
      COMMON /MULREL/ PLHEAT(4), PLHITE(4), PLUDUR(4),
     $                PDELAY(4), REFTIM(4), RELINV(150,4), PSDIST(10,10)

      DIMENSION AO(720)

      EQUIVALENCE (AO,PLHEAT)

      DATA AO/720*0./

      END
```

A.2 Compiler Options

The program was compiled using Version 3.01 of the Lahey F77L-EM/32 FORTRAN-77 compiler with the options shown below.

Option	Description
/n0	- Standard FORTRAN-77 IMPLICIT
/n7	- Optimize inter-statement
/nB	- No Bounds checking
/nC	- Ignore nonstandard usage
/nD	- DIRECT files with headers
/nH	- No Hardcopy source listing
/nI	- No Interface checking
/nK	- Generate 80x87 code
/ L	- Line-number traceback table
/ P	- Protect constant arguments
/nQ	- No Quirky situations
/ R	- Remember local variables
/ S	- Create filename.SLD for SOLD
/nT	- INTEGER*4, LOGICAL*4 default
/ W	- Display Warning messages
/nX	- No Xref listing
/nZ1	- Better SOLD debugging

DISTRIBUTION

U.S. NUCLEAR REGULATORY COMMISSION (39)

WILLIAM BECKNER, NRR/PRAB, MS 10A2

FRANK CONGEL, NRR/DREP, MS 10E4

DONALD COOL, RES/RPHEB, MS NLS-139

LEMOINE CUNNINGHAM, NRR/PRAB, MS 11D23

MARK CUNNINGHAM, RES/DSR/PRAB, MS NLS-372

A. BERT DAVIS, REGION III

STEWART EBNETER, REGION II

JAMES GLYNN, RES/PRAB, MS NLS-372

MICHAEL JAMGOCHIAN, RES/SAIB, MS NLS-324

JAMES JOHNSON, RES/PRAB, MS NLS-372

THOMAS KING, RES/DSIR, MS NLS-360

ROBERT KORNASIEWICZ, RES/WMB, MS NLS-260

LES LANCASTER, RES/PRAB, MS NLS-372

JOE LEVINE, NRR/PRPB, MS 11D23

TIM MARGULIES, RES/WMB, MS NLS-260

JOHN MARTIN, REGION V

ROBERT MARTIN, REGION IV

TOM MCKENNA, AEOD/IRB, MS 3206

WARREN MINNERS, RES/DSIR, MS NL-007

JOCELYN MITCHELL, RES/SAIB, MS NLS-324

BILL MORRIS, RES/DRA, MS NL-007

JOSEPH MURPHY, RES/DSR, MS NL-007

ROBERT PALLA, NRR/PRAB, MS 10A2

HAROLD PETERSON, RES/RPHEB, MS NLS-139

EDWARD PODOLAK, NRR/PEPB, MS 10D4

JOHN RIDGELY, RES/SAIB, MS NLS-324

DENWOOD ROSS, AEOD, MS 3701

ZOLTAN ROSZTOCZY, AEOD, MS 3701

WILLIAM RUSSELL, REGION I

CHRIS RYDER, RES/PRAB, MS NLS-372

BRIAN SHERON, RES/DSR, MS NL-007

FRANK SKOPEC, NRR/PRPB, MS 11D23

LEONARD SOFFER, RES/SAIB, MS NLS-324

THEMIS SPEIS, RES, MS NL-007

MAT TAYLOR, NRC/EDO, MS 17G21

ASHOK THADANI, NRR/DST, MS 8E2

HAROLD VANDERMOLEN, RES/PRAB, MS NLS-372

CHARLES WILLIS, NRR/DREP, MS 10E4

SHLOMO YANIV, RES/RPHEB, MS NLS-139

U.S. ENVIRONMENTAL PROTECTION AGENCY (2)
OFFICE OF RADIATION PROGRAMS
Attn: JOE LOGSDON
ALLEN RICHARDSON
WASHINGTON, DC 20460

BRUCE BURNETT, CDRH (HFZ-60)
U.S. DEPT. OF HEALTH & HUMAN SERVICES
FOOD AND DRUG ADMINISTRATION
5600 FISHERS LANE
ROCKVILLE, MD 20857

SARBES ACHARYA, MS NS-20
U.S. DEPARTMENT OF ENERGY
19901 GERMANTOWN ROAD
GERMANTOWN, MD 20585

U.S. DEPARTMENT OF ENERGY (2)
Attn: ED BRANAGAN, MS EH352
KEN MURPHY, MS EH351
WASHINGTON, DC 20545

ROBERT OSTMEYER
ROCKY FLATS AREA OFFICE
U.S. DEPARTMENT OF ENERGY
P.O. BOX 928
GOLDEN, CO 80402

DAVID BLACK
AMERICAN ELECTRIC POWER
1 RIVERSIDE PLAZA
COLUMBUS, OH 43215

NATIONAL ENERGY SOFTWARE CENTER (20)
ARGONNE NATIONAL LABORATORY
Attn: LARRY EYBERGER
9700 SOUTH CASS AVENUE
ARGONNE, IL 60439

ARGONNE NATIONAL LABORATORY (3)
Attn: S. Y. CHEN
KOU-JOHN HONG
BRAD MICKLICH
9700 SOUTH CASS AVENUE
ARGONNE, IL 60439

JON HELTON
MATHEMATICS DEPARTMENT
ARIZONA STATE UNIVERSITY
TEMPE, AZ 85287

WILLIAM HOPKINS
BECHTEL POWER CORPORATION
15740 SHADY GROVE ROAD
GAITHERSBURG, MD 20877

BROOKHAVEN NATIONAL LABORATORY (2)
Attn: CARRIE GRIMSHAW
ARTHUR TINGLE
BUILDING 130
UPTON, NY 11973

ROBERT GOBEL
CLARK UNIVERSITY
950 MAIN STREET
WORCESTER, MA 01610

DUNCAN BREWER
DUKE POWER CO.
422 SOUTH CHURCH STREET
CHARLOTTE, NC 28242

JIM MAYBERRY
EBASCO SERVICES
160 CHUBB AVENUE
LYNDHURST, NJ 07071

EG&G IDAHO, INC. (5)
Attn: R. J. DALLMAN, MS 2508
C. A. DOBBE, MS 2508
K. R. JONES, MS 2508
D. L. KNUDSON, MS 2508
R. E. LYON, MS 3523
P.O. BOX 1625
IDAHO FALLS, ID 83415

TERRY FOPPE
ENERGY SYSTEMS GROUP
EG&G ROCKY FLATS, INC.
P.O. BOX 464
GOLDEN, CO 80401

IAN WALL
ELECTRIC POWER RESEARCH INSTITUTE
3412 HILLVIEW AVE.
PALO ALTO, CA 94304

GERALD DAVIDSON
FAUSKE AND ASSOCIATES, INC.
16 W 070 WEST 83RD STREET
BURR RIDGE, IL 60521

JOHN LUKE
FLORIDA POWER & LIGHT
P.O. BOX 14000
JUNO BEACH, FL 33408

CHRISTINE MILLER
F-30
KOSHLAND WAY
SANTA CRUZ, CA 95064

HAROLD CAREWAY, M/C 754
GENERAL ELECTRIC CO.
175 CURTNER AVENUE
SAN JOSE, CA 95129

JUDY ROLLSTIN
GRAM, INC.
1709 MOON NE
ALBUQUERQUE, NM 87112

KNOLLS ATOMIC POWER LABORATORY (2)
Attn: KEN MCDONOUGH
DOMINIC SCIAUDONE
BOX 1072
SCHENECTADY, NY 12301

LAWRENCE LIVERMORE NATIONAL LABORATORY (3)
Attn: MARVIN DICKERSON
GEORGE GREENLY
ROLF LANGE
LIVERMORE, CA 94550

LOS ALAMOS NATIONAL LABORATORY (2)
ANALYSIS AND ASSESSMENT DIVISION
Attn: JANE BOOKER
MARY MEYER
LOS ALAMOS, NM 87545

BILL EAKIN
NORTHEAST UTILITIES
BOX 270
HARTFORD, CT 06141

RAY NG
NUMARC
SUITE 300
1776 EYE STREET, NW
WASHINGTON, DC 20006

MIKE CHEOK
NUS
910 CLOPPER ROAD
GAITHERSBURG, MD 20878

OAK RIDGE NATIONAL LABORATORY (2)
Attn: KEITH ECKERMAN
ROBERT ROUSSIN
P.O. BOX 2008
OAK RIDGE, TN 37831

DENNIS STRENGE, RTO/125
PACIFIC NORTHWEST LABORATORY
P.O. BOX 999
RICHLAND, WA 99352

R. TOOSSI
PHYSICAL RESEARCH, INC.
25500 HAWTHORNE BLVD.
TORRANCE, CA 90505

KEITH WOODARD
PICKARD, LOWE AND GARRICK
SUITE 730
1615 M STREET, NW
WASHINGTON, DC 20036

ZEN MENDOZA
SAIC
SUITE C31
5150 EL CAMINO REAL
LOS ALTOS, CA 94022

SAIC (2)
Attn: CHRIS AMOS
PAUL MATTINGLY
2109 AIR PARK RD., SE
ALBUQUERQUE, NM 87106

SAIC (3)
Attn: DAVE ALDRICH
ROGER BLOND
GEOFF KAISER
MAIL STOP 2-5-1
1710 GOODRIDGE DRIVE
MCLEAN, VA 22102

FRED HARPER
DIVISION 6412
SANDIA NATIONAL LABORATORIES
ALBUQUERQUE, NM 87185

SAVANNAH RIVER LABORATORY (2)
Attn: KEVIN O'KULA
DAVE SHARP
AIKEN, SC 29808

JIM MEYER
SUITE 100
SCIENTECH
11821 PARKLAWN DRIVE
ROCKVILLE, MD 20852

SCOTT BIGELOW
S-CUBED
SUITE 300
2501 YEALE SE
ALBUQUERQUE, NM 87106

EDWARD WARMAN
STONE & WEBSTER ENGINEERING CORP.
P.O. BOX 2325
BOSTON, MA 02107

TECHNADYNE ENGINEERING CONSULTANTS (3)
Attn: DAVID CHANIN
BURT NEWMARK
MEL PIEPHO
P.O. BOX 13928
ALBUQUERQUE, NM 87192

KEN KEITH
TENNESSEE VALLEY AUTHORITY
W 10 D 201
400 WEST SUMMIT HILL
KNOXVILLE, TN 37902

F. ERIC HASKIN
DEPT. OF NUCLEAR ENGINEERING
UNIVERSITY OF NEW MEXICO
ALBUQUERQUE, NM 87131

KEN O'BRIEN
UNIVERSITY OF WISCONSIN
153 ENGINEERING RESEARCH BLDG.
MADISON, WI 53706

GRIFF HOLMES
WESTINGHOUSE ELECTRIC CORP.
ENERGY CENTER EAST, BLDG. 371
P.O. BOX 355
PITTSBURGH, PA 15230

FRED MANN, W/A-53
WESTINGHOUSE HANFORD CO.
P.O. BOX 1970
RICHLAND, WA 99352

SHENGDAR LEE
YANKEE ATOMIC ELECTRIC COMPANY
580 MAIN STREET
BOSTON, MA 01740

EPHRAIM ASCULAI
DIVISION OF NUCLEAR SAFETY
WAGRAMESTRASSE, 5
P.O. BOX 100
A-1400 WIEN
AUSTRIA

PAUL GOVAERTS
DEPARTMENT DE LA SURETTE NUCLEAIRE
AVENUE DU ROI 157
B 1060 BRUXELLES
BELGIUM

G. NEALE KELLY
NUCLEAR SAFETY RESEARCH
COMMISSION OF THE EUROPEAN COMMUNITIES
RUE DE LA LOI, 200
B-1049 BRUXELLES
BELGIUM

NADIA SOIDO FALCAO MARTINS
COMISSAO NACIONAL DE ENERGIA NUCLEAR
R GENRAL SEVERIANO 90 S/408-A
RIO DE JANEIRO
BRAZIL

S. DAGGUPATY
ENVIRONMENT CANADA
4905 DUFFERIN STREET
DOWNSVIEW
ONTARIO, M3H 5T4
CANADA

SOREN THYKIER-NIELSEN
RISO NATIONAL LABORATORY
POSTBOX 49
DK-4000 ROSKILDE
DENMARK

STEPHEN BOULT
ELECTROWATT ENGINEERING SERVICES (UK) LTD.
GRANDFORD HOUSE
16 CARFAX, HORSHAM
WEST SUSSEX RH12 1UP
ENGLAND

SEPPO VUORI
TECHNICAL RESEARCH CENTRE OF FINLAND (VTT)
NUCLEAR ENGINEERING LABORATORY (YDI)
LONNROTINKATU 37
P.O. BOX 169
SF-00181 HELSINKI 18
FINLAND

DANIEL MANESSE
IPSN
BOITE POSTALE 6
F-92265 FONTENAY-AUX-ROSES CEDEX
FRANCE

JOACHIM EHRHARDT
INSTITUT FUR NEUTRONENPHYSIK UND
REAKTORTECHNIK (INR)
KERNFORSCHUNGSZENTRUM KARLSRUHE GMBH
POSTFACH 3640
D-7500 KARLSRUHE 1
FEDERAL REPUBLIC OF GERMANY

JOHN G. KOLLAS
INSTITUTE OF NUCLEAR TECHNOLOGY & RADIATION
PROTECTION
N.R.C.P.S. "DEMOKRITOS"
P.O. BOX 60228
GR-153 10 AGHIA PARASKEVI
ATTIKI
GREECE

M. K. YEUNG
MECHANICAL ENGINEERING DEPT.
UNIVERSITY OF HONG KONG
POKFULAM
HONG KONG

ELI STERN
ISRAEL AEC LICENSING DIV.
P.O. BOX 7061
TEL-AVIV 61070
ISRAEL

ENEA/DISP (2)
Attn: ALFREDO BOTTINO
ALVARO VALERI
VIA VITALIANO BRANCATI, 48
00144 ROMA EUR
ITALY

HIDEO MATSUZURU
TOKAI RESEARCH ESTABLISHMENT
TOKAI-MURA
MAKA-GUN
IBARAKI-KEN, 319-11
JAPAN

PAUL KAYSER
DIVISION DE LA RADIOPROTECTION
1, AVENUE DES ARCHIDUCS
L-1135 LUXEMBOURG-BELAIR
LUXEMBOURG

JAN VAN DER STEEN
KEMA LABORATORIES
UTRECHTSEWEG, 310
POSTBUS 9035
NL-6800 ET ARNHEM
NETHERLANDS

ULF TVETEN, HEAD
ENVIRONMENTAL PHYSICS SECTION
INSTITUTT FOR ENERGITEKNIKK
POSTBOKS 40
N-2007 KJELLER
NORWAY

LEONEL CANELAS
NEW UNIVERSITY OF LISBON
QUINTA DE TORRE
2825 MONTE DA CAPARICA
PORTUGAL

D. EUGENIO GIL LOPEZ
CONSEJO DE SEGURIDAD NUCLEAR
CALLE JUSTO DORADO, 11
E-28040 MADRID
SPAIN

LENNART DEVELL
STUDSVIK NUCLEAR
STUDSVIK ENERGITEKNIK AB
S-611 82 NYKOPING
SWEDEN

HANSPETER ISAAK
ABTEILUNG STRAHLENSCHUTZ
HAUPTABTEILUNG FUR DIE SICHERHEIT
DER KERNANLAGEN (HSK)
CH-5303 WURENLINGEN
SWITZERLAND

DER-YU HSIA
ATOMIC ENERGY COUNCIL
67, LANE 144
KEELUNG ROAD, SECTION 4
TAIPEI, TAIWAN 10772

MARION HILL
NATIONAL RADIOLOGICAL PROTECTION BOARD
CHILTON
DIDCOT
OXON. OX11 ORQ
UNITED KINGDOM

SHANKARAN NAIR
CENTRAL ELECTRICITY GENERATING BOARD
BERKELEY NUCLEAR LABORATORIES
BERKELEY
GLOUCESTERSHIRE GL13 9PB
UNITED KINGDOM

WILLIAM NIXON
AEA/SRD
WIGSHAW LANE
CULCHETH
WARRINGTON
CHESHIRE WA3 4NE
UNITED KINGDOM