CONTRACT REPORT ARBRL-CR-00466

SAIL USER GUIDE FOR RUNNING THE HULL AND EPIC3 CODES

Prepared by

Orlando Technology, Incorporated
P. O. Box 855
Shalimar, FL 32579

September 1981

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Destroy this report when it is no longer needed.
Do not return it to the originator.

Secondary distribution of this report by originating or sponsoring activity is prohibited.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22151.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Date Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: CONTRACT REPORT ARBRL-CR-00466

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*: SAIL User Guide for Running the HULL and EPIC3 Codes

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): Daniel A. Matuska

8. CONTRACT OR GRANT NUMBER(s): DAAK11-79-C-0106

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Orlando Technology, Incorporated
P.O. Box 855
Shalimar, Florida 32579

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS:
US Army Armament Research and Development Command
US Army Ballistic Research Laboratory
DRDAR-BL
Aberdeen Proving Ground, MD 21005

12. REPORT DATE: September 1981

13. NUMBER OF PAGES: 38

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*

15. SECURITY CLASS. *(of this report)*: UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

AFATL-TR-81-70

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Software Management
User Software
Executive Processor
Data Management

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report serves as documentation of the SAIL executive processor system. SAIL is a user-oriented computer program which was designed to manage large files of FORTRAN code to facilitate code development and management. The use of SAIL permits automatic selection and expansion of the FORTRAN source in an attempt to reduce core and central processor requirements. SAIL is written in FORTRAN for use on CDC, IBM, Honeywell and DEC computers.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Date Entered)*

FORWARD

This report documents the SAIL code as implemented to support the HULL and EPIC3 continuum mechanics computer codes at the Army Ballistic Research Laboratory, the Air Force Armament Laboratory, and the Air Force Weapons Laboratory. This work was performed by Daniel A. Matuska, Orlando Technology, Incorporated, during the period August 1979 through September 1980. The BRL Project Manager for this effort was Dr. John Zukas.

ACKNOWLEDGEMENTS

The original SAIL code was developed from concepts borrowed from Dr. Reginald W. Clemens as implemented by the author and Richard E. Durrett during the initial development of the HULL code in 1971. These concepts were expanded to include an enlarged syntax and were combined with an update system by Lewis P. Gaby in December of 1973. Since that time, SAIL has been continually exercised and debugged. The resulting product is due to a community of government and contractor personnel too numerous to mention.

TABLE OF CONTENTS

LIST OF TABLES

SECTION I
INTRODUCTION

The initial design and development of the HULL finite difference hydrodynamics computer code in 1971 (Reference 1) was done in the framework of a simple executive pre-processor. The function of this executive was to manipulate pre-existing FORTRAN code to minimize central processor time, alter the size of dimensioned arrays to fit the particular HULL problem being run to minimize occupancy of central memory, and to permit inclusion of redundant statements (i.e., COMMON Variable definitions) only once in order to reduce the possibility of coding and keypunch errors. The previous experience of the HULL code authors with other large finite-difference codes had proved that no matter how flexible a code was when designed, the advent of new problems, computational techniques and user demand would invariably lead to the production of new and usually divergent versions of the same code. Since the HULL code was to be an experimental test bed for developing numerical techniques and for solving non-linear physics problems which could not be addressed by other codes, the executive pre-processor was given the capability to selectively retain coded segments for subsequent inclusion based on user selected option fields. Thus if a new feature was to be added for a particular application, the user/coder could invent a descriptive option name, contain the added code within the option selecting syntax, and leave the remainder of the code functioning as before. Once the added feature was checked out and successfully run, it became a permanent option available for use by other HULL code users.

During this early development stage, the CDC UPDATE utility did not exist and the IBM IEBUPDATE utility was subject to constant change and therefore prone to execution error. Consequently the local AFWL update system was used to maintain the HULL system file. This update utility could perform deletions, changes, and additions to a file, but would produce a fatal error if statements were altered out-of-sequence or if its format requirements were not strictly observed.

The first version of SAIL (Reference 2) duplicated the functions of the AFWL update system while simplifying the nomenclature. Later versions of SAIL incorporated all of the executive pre-processor functions. This allowed the user to incorporate changes and produce compiler input in a single pass by executing program SAIL.

From 1973 to the present, the main advantage of SAIL has been that it has been under the control of the users. Functions have been added only when a clear need existed. Maintenance of SAIL is required when operating systems are changed but this is usually not too traumatic as most of SAIL is written in relatively machine independent FORTRAN.

SAIL was never intended to replace or supplement existing system software utilities. It was intended to serve as a user oriented tool for maintenance and development of the HULL code. The development of SAIL has led to the definition of a simple fairly-well structured high level language that can be used to assemble FORTRAN source code in a manner analagous to a compiler producing assembler instructions. The SAIL code was not intended to be a black box whose functions are transparent to the user. The fact that SAIL has become widely used for applications other than the HULL code is a testament to its usefulness, since its acquisition and installation is not trivial. The first documentation for installation and use of SAIL in a more global sense did not appear until 1979 (Reference 3). Since that time maintenance of SAIL has been shared by the Air Force Weapons Laboratory (AFWL/NTYP) and the Air Force Armament Laboratory (AFATL/DLJW). At present SAIL is being used to manage the HULL code at several other installations within the United States and the United Kingdom.

The purpose of this document is to guide users in managing and assembling FORTRAN source code for EPIC3 and HULL by using SAIL.

SECTION II
SAIL FILE STRUCTURE

The SAIL code operates on a file which is in a specific format to simplify processing and reduce input/output time. This file is produced by SAIL in either the generate, copy, convert or update mode. This will be referred to as the library file.

The first record of the library file is a header containing the information indicated by Table 1.

TABLE 1. SAIL LIBRARY FILE - RECORD 1 CONTENTS FOR RECORD LENGTH N

WORD	CONTENTS	TYPE
1	Version number	(Integer)
2	System Name	(Alpha)
3	Creation Date	(Alpha)
4	Number of default programs	(Integer)
5	First default program Name	(Alpha)
6	Second default program Name	(Alpha)
.		
.		
.		
50	Forty sixth default program Name	(Alpha)
51	Name of first program on this file	(Alpha)
52	Sequence number of first program start	(Integer)
53	Name of second program on this file	(Alpha)
54	Sequence number of second program start	(Integer)
.		
.		
.		
N-1	Name of (N-50)/2 program on this file	(Alpha)
N	Sequence number of (N-50)/2 program start	(Integer)

The system name and version are identifiers which may be arbitrary. The version number is automatically incremented by 1 during a SAIL update if no other action is taken. The creation date is the date that the current version was produced. The default programs are the names of programs that will be assembled by SAIL in the Executive mode. The body of each SAIL file can be

subdivided into separate programs. Each program is assigned a sequence number which is a multiple of 10,000; therefore the first program would start at sequence number 10,000, the second at 20,000 if the first program is less than 10,000 statements in length or at the next multiple of 10,000 greater than the length of program one plus 10,000. If a collection of statements is used in more than one program, these statements can be placed ahead of the first program starting at sequence number 1. This portion of the file is called the Prologue and is always processed by SAIL during the Executive mode.

The second record on the SAIL library (immediately preceding the prologue or the first program) contains the information described by Table 2. The number of cards or statements on this file is for information only and is not used by SAIL to process the file. The option names in Record 2 are referred to as the option directory. Their purpose is to define option names and to assign corresponding integer values to be used in controlling the processing by SAIL during the Executive mode. Option directory entries can be changed, deleted or added only during generate, copy or update mode.

TABLE 2. SAIL LIBRARY FILE - RECORD 2 CONTENTS FOR RECORD LENGTH N

WORD	CONTENTS	TYPE
1	Number of Cards on this file	(Integer)
2	Number of default options	(Integer)
3	Name of Option 1	(Alpha)
4	Value assigned Option 1	(Integer)
5	Name of Option 2	(Alpha)
6	Value assigned Option 2	(Integer)
.		
.		
.		
N-1	Name of Option (N-2)/2	(Alpha)
N	Value assigned Option (N-2)/2	

The remainder of the file following Record 2 consists of data which will be used by SAIL in assembling input for the compiler during the SAIL Executive mode. These data are organized into groups of individually sequenced statements so that an input/output buffer operation can treat a large amount of

data during each call. On CDC machines SAIL defaults to 113 sequenced statements or FORTRAN card images per buffer group. Each statement is composed of a single word sequence number followed by 72 characters of data. Each statement is terminated by 8 characters which contain the date of insertion or last modification of that sequence number.

The individually sequenced statements can be thought of as 72 column card images. These data are of three forms:

(1) SAIL informative directives or aids to simplify file management.

(2) SAIL Executive directives which control and modify assembly of the compiler input card images.

(3) Compiler input data kernels to be output directly or in modified form after selection by SAIL Executive directives.

SECTION III
SAIL DIRECTIVES

There are three basic types of SAIL directives. The distinction between them blurs at times because some directives serve multiple functions and can belong to more than one type class. All directive types are input to SAIL in free format with the exception that all directives must start in column one or its equivalent on the input file. The free format delimiters for SAIL are the blank field, comma, or the equal symbol.

1. SAIL Executive Directives

SAIL Executive or assembly execution directives consist of up to three basic fields. The fields are separated by the SAIL delimiters and are order dependent. The first field is the verb field, the second is the noun or subject field, and the third field, when present, is the operand or option field. An exhaustive discussion of the attributes, of each field is given by Gaby et al (Reference 3) and will be briefly summarized here for completeness. All Executive directives are contained within the body of the SAIL library file to direct processing during executive assembly. All of these directives are preceded by an asterisk which must appear as the first character in the sequenced statement. Directives are not included in the compiler input file produced by SAIL during assembly execution. The current set of executive directives is listed in Table 3.

TABLE 3. SAIL EXECUTIVE DIRECTIVES

VERB	NOUN		OPERAND	
*AUTO				
*B	PROGRAM NAME			
*DEFL	OPTION NAME	LOGICAL	OPTION	STRING
*DEFN	OPTION NAME	NUMERIC	OPTION	STRING
*E				
*ENDPROC				
*ETXT				
*INCLUDE	PROC NAME	LOGICAL	OPTION	STRING
*KEEPTO	LABEL NAME	LOGICAL	OPTION	STRING
*LABEL	LABEL NAME			
*PROC	PROC NAME	LOGICAL	OPTION	STRING
*SKIPTO	LABEL NAME	LOGICAL	OPTION	STRING
*TXT				

A. Directive Verb Field

Executive processing directives start with an asterisk in the first character of the record (i.e. in column one of a card image). The alphabetical characters which follow the asterisk in the verb field must be adjacent to one another (no embedded blanks). The Executive directives and Change directives will not appear in the resulting executive compiler input file. If a text string is not recognized as a directive, it will be included in the executive compiler input file. Since it has an asterisk in column one, it will be recognized as a comment card by most FORTRAN compilers.

B. Directive Noun Field

Noun fields, as indicated by Table 3, are not used by all Executive directives. When present they consist of up to eight non-blank characters which are left to the discretion of the user. In general it is best to limit the definition of noun fields to alphabetic characters. Special forms of the noun field are employed by the *PROC and *INCLUDE directives. These exceptions will be discussed later.

C. Directive Operands

Operands are either logical or numeric. Only the *DEFN directive uses the numeric operand form. Numeric operands consist of option names and integer constants separated by the arithmetic operators +, -, /(division), and *(multiplication). The order of evaluation is strictly from left to right. The result of evaluation is always non-negative. Logical operands have only true/false values. The value true is indicated by a numeric value of greater than zero, false by a value of zero. Logical operands are formed by separating option names and integer constants by the logical relations EQ, NE, LE, LT, GE, GT. Logical operands can also be assembled by connecting options or groups of logical relations by the logical operators AND, OR, and NOT. When integer constants are used in conjunction with a relational operator, the constant must follow the relational operator without delimiters (blank, comma, equal). Relational operators and option names must be separated by blanks. The use of undefined option names will result in abnormal termination. A logical test for the existence of an option name can be performed by using the special relational operator "DEF" followed by an option name. If the option exists (with zero or integer value) the element has a value of true. If the option has not been previously defined, the element has a value of false. As in the

case of numeric operands, logical operands are evaluated from left to right. Groups of relational elements may appear between parentheses, in which case the parenthetical groups will be evaluated first.

D. Option Definition and Redefinition

Options can be initially defined in one of three ways; by the SAIL file option directory record, by *DEFN or *DEFL Executive directives, or by the SAIL input stream. The option directory was previously described. Its initial definition will be discussed later under Maintenance directives. The SAIL input conventions will be defined in the INPUT/OUTPUT section of this document. If an option has been previously defined by either the SAIL file directory or the input data stream, its value can be changed by SAIL during Executive processing by:

```
*DEFN Name Arithmetic Operand
```

in which case the option Name will have the new value produced by evaluation of the Arithmetic operand. Options can also be changed by:

```
*DEFL Name Logical Operand
```

which will result in option Name having a value of zero if the result of evaluating the Logical operand is false, and a value of one if the operand result is true. Both the *DEFN and *DEFL directive can be used to define new options if they have not been previously defined. As an example, assume the option directory contained the options and values:

```
A=1
B=2
C=0
D=10
E=22
```

SAIL execution of

```
*DEFN D A+E*C
```

would produce

D=0, since numeric operands are performed from left to right.

```
*DEFN NEW E+B
```

would produce

NEW=24.

*DEFL A B EQ1

would produce

A=0 since option B is not equal to one.

*DEFL H "DEF" X

would produce

H=0, since option x was not previously defined.

*DEFL G D

would produce

G=1, since option D is true (non zero).

E. SAIL Dynamic File Modifications

SAIL can modify the compiler input stream by using the values of options which have been defined statically by the option directory or dynamically by the input data stream and subsequent *DEFL and *DEFN directives. Segments of the file can be included, left out, or replicated. Individually sequenced card images can be altered by value replacement or new images can be produced. All of these functions can be controlled by the values of options when used in conjunction with the SAIL Executive directives *PROC, *INCLUDE, *SKIPTO, and dynamic value substitution.

(1) *PROC and *INCLUDE

The procedure definition or *PROC construct was the first Executive syntactical element used in the HULL code. Its basic form is:

```
*PROC NAME Logical Operand
S1
S2
.
.
.
SN
*ENDPROC
```

The collection of statements S1 through SN between the *PROC - *ENDPROC pair will be stored with the descriptor NAME if the logical operand has a value of true. The statements S1 through SN can be any Executive directive except another *PROC definition. The data and directives within the procedure will be expanded by SAIL prior to storing the compile file. If the operand field is blank, the procedure will be unconditionally created - that is, it will always be stored with the procedure definition of NAME.

This procedure is invoked by the directive:

*INCLUDE NAME logical operand

at the point in the SAIL file where it appears if the logical operand has a value of true. If the procedure NAME has not been defined, a message will be printed to output and SAIL will abort. Procedures can be included by the INCLUDE directive within the definition of other procedures but only up to a nesting depth of eight. If the *ENDPROC directive is not found before the end of file or before encountering another *PROC directive, abnormal termination will result. If the logical operand field is blank, the procedure will be unconditionally included.

A second form of the *PROC construct permits the definition of a macro-like feature. In this form the noun field is concatenated with a series of arguments enclosed in parentheses. Thus we might define:

```
*PROC SQRT(A,B,C)
"A"=SQRT("B"**2-4*"A"*"C")
*ENDPROC
```

where the arguments enclosed in " " in the procedure statement are to be replaced upon expansion. To invoke the procedure defined above the Executive directive:

*INCLUDE SQRT("X","Y","Z")

would result in

X=SQRT(Y**2-4*X*Z).

Notice that in these examples of procedure definition and procedure inclusion, the operand field is blank. This is an unconditional form of definition that is simple but may require more execution time and I/O overhead if the procedure definitions are made but not subsequently invoked by *INCLUDE directives. Finally a third form of procedure definition is possible by using a table entry to define a character string for subsequent use in

building a procedure name for inclusion dependent on the table contents. In this usage the noun or name field of the *INCLUDE directive is delimited by the character $ in the following form:

```
*INCLUDE $ NAMENN AAA $ logical operand.
```

where NAME is the name of a previously defined option table and NN is the value of an option in the table. The field designated by AAA is not required but, if present, will be concatenated with the table entry option name to produce the procedure name to be included. If the table entry does not exist, SAIL ignores the directive. If the table entry does exist but the resulting procedure name has not been defined, SAIL will terminate abnormally. As an example of this usage, assume the following options are defined with the values indicated:

```
NM=3
EOS=6
MAT=3
AIR=1
AL=3
FE=2
NH=22
XX=30
```

This construct is used by HULL to generate the equations of state to be used in multi-material calculations. This is done by:

```
*PROC          AIREOS          "DEF" AIR
  S1
*ENDPROC

*PROC          ALEOS           "DEF" AL
  S2
*ENDPROC

*PROC          CUEOS           "DEF" CU
  S3
*ENDPROC
*PROC          FEEOS           "DEF" FE
  S4
*ENDPROC

*PROC          NIEOS           "DEF" NI
  S5
*ENDPROC
```

```
.
.
.
.
*INCLUDE $ MAT1 EOS $ NM

*INCLUDE $ MAT2 EOS $ NM

*INCLUDE $ MAT3 EOS $ NM

*INCLUDE $ MAT4 EOS $ NM.
```

The statements S1 through S5 represent the equation of state routines for each of the materials referenced by the noun and operand fields. Recall that procedures are not defined unless the referenced option names have been previously defined. Thus, the NIEOS and CUEOS procedures are not defined in the above example since the NI and CU options were not previously defined. The option table or directory assumed for this example contains the names of three recognizeable material names. They appear immediately after the option MAT which has a value of three. Thus the order of the existing option directory determines the definition of a table. Options AIR, AL and FE are the sole members of table MAT. Note that the existence of other tables is implied by the order and numeric values of the options. Thus table NM consists of the three elements EOS, MAT, and AIR and table EOS has the six elements MAT, AIR, AL, FE, NH, and XX. The procedure names are invoked in this case by the example *INCLUDE directives which will be initially expanded to:

```
*INCLUDE        AIREOS        NM
*INCLUDE        FEEOS         NM
*INCLUDE        ALEOS         NM
```

The last *INCLUDE directive is not considered since no option value of 4 was found within the range of table MAT. The final result produced for this sequence of *PROC definitions and the corresponding *INCLUDE directives will be insertion on the compilable output of the three program segments:

```
S1
S4
S2
```

which represent the equation of state routines for Air, FE(Iron) and AL(Aluminium) respectively.

(2) ***SKIPTO, *KEEPTO, and *LABEL**

The *SKIPTO/*KEEPTO directives are logical complements of one another. For small numbers of card images the form is:

*KEEPTO *N logical operand.

This directive will cause SAIL to retain or keep the next N card images in the library file if the evaluated operand is true, otherwise the next N cards will be skipped or left out of the compilable file. The same result could be achieved by replacing the *KEEPTO verb with the *SKIPTO verb and preceding the logical operand by the logical operator NOT. When N is larger than five the effect of the *SKIPTO/*KEEPTO construct becomes difficult to read on a master SAIL library file listing and it is clearer to use the alternate form:

```
*SKIPTO          label name          logical operand
   C1
   C2
.
.
.
   CN
*LABEL           label name
```

which would result in card images C1-CN being bypassed if the value of the logical operand is true. The noun field, label name, can be any eight alphanumeric characters. The resulting label names are easier to read if the first character is alphabetic. The sequenced card images that appear with the range of a *SKIPTO/*KEEPTO may be of any type including SAIL Executive directives. SAIL comments (statements with the character = in column one), *P, *ETXT, *DIR and *EDIR are not counted within the range of a *SKIPTO/*KEEPTO with a noun field of the form *N. *SKIPTO/*KEEPTO-*LABEL groups can be nested using the same conventions as the FORTRAN DO loop. For example, assume a set of options and respective values are:

```
NSTN=0
ATMOS=5
NOP=100.
```

Then the Executive directives:

```
*KEEPTO *1 NSTN
   C1

*SKIPTO END1 NOP GE200
```

```
   C2
   C3
   C4
*LABEL END1

*KEEPTO END2 NOP GE200
   C5
   C6
   C7
   C8
*LABEL END2
```

will result in the retention of the card images:

```
C2
C3
C4.
```

Care must be exercised in the use of the *SKIPTO/*KEEPTO construct lest a *PROC or *ENDPROC directive be left out and a procedure definition be incorrectly established.

(3) DYNAMIC VALUE AND STRING SUBSTITUTION

Dynamic substitution is performed only on those card images which have the character $ in column one or are preceded by and followed by the directives *AUTO and *MAN. For example:

```
*AUTO
C1
*MAN.
```

Substitution of previously defined options by their current values is made for those option names delimited by the characters (,), and /. These delimiters can appear in the usual mixed fashion as they would in the case of FORTRAN DIMENSION and DATA declarations. Thus if the option name/value pairs:

```
NH=16
IMAX=60
JMAX=100
```

are assumed, then the statements:

```
$  COMMON/EOS/RCSQ(IMAX,JMAX)
$  DATA NH/NH/
```

would result in:

```
COMMON/EOS/RCSQ(60,100)
DATA NH/16/.
```

Note that the extraneous symbols ($) are removed. If the option is not defined then the card image will be processed without removal of the character string and SAIL will continue normal processing. If it is desired to dynamically substitute a value which would not conform to the above construct, then the form:

```
$  ABCE_OPTION NAME_EFG
```

can be used. The option whose value is to be substituted is delimited by the character _. String substitution can be performed by use of the option table construct further delimited by a preceding $ symbol. This is of the form:

```
$  ABCD_$ NAMENN_EFGH
```

where the option table name is as discussed previously under *PROC directives and NN is the value of the option in the table that will be inserted. Assume the existence of option names/values:

```
NM=3
FE=4
AL=1
AIR=6
```

The following statements:

```
*AUTO
DO 100 I=1, _NM+2_
*MAN
$ CALL _$NM6_(P,_NM_)
```

would result in:

```
DO 100 I=1,5
CALL AIR(P,3).
```

F. Program Definition and Termination

The SAIL Executive directives:

```
*B Program name
   P1
*E
```

are used to delimit a program segment. Thus if processing of a specific program on a SAIL library is requested the SAIL code will process the PROLOGUE and skip to the program name desired as defined by the *B directive. Processing will terminate upon encountering an *E or a subsequent *B directive. In addition, a SAIL library listing will include the program name specified by *B directives in the summary and will force the listing of that beginning of program name to start at the top of a page with the program name appearing in the banner.

G. *TXT and *ETXT Directives

SAIL will process all recognizeable Executive directives during the Executive assembly process. There are situations in which these directives are used as data. Directives for which SAIL Executive processing is to be suppressed can be contained between *TXT/*ETXT pairs. Therefore the segment:

```
*TXT
*KEEPTO *1 NOP
  C1

*PROC AIR NM EQ1
  C2

*ENDPROC

*ETXT
```

would be preserved on the compileable output file exactly as they appear above except the *TXT and *ETXT directives are removed.

H. SAIL Change Directives

After a SAIL library file has been established or updated, it may be necessary to change, add, or delete statements either in the course of Executive assembly of a compiler input file or to produce an updated version of the library file.

The allowable Change directives are:

```
*A N1
*I N1
*C N1,N2
*C N1,(C1,C2,NC1,NC2)
*D N1,N2
*M N1
```

where N1 and N2 are the first and last sequenced card images affected by the directives. Records are inserted by:

```
*A N1
*I N1.
```

All cards between these directives on the SAIL input file and a subsequent Change directive or input file end will be inserted after sequence number N1. Records are deleted by:

```
*C N1,N2
*D N1,N2.
```

All cards with sequence numbers N1 through N2 will be deleted from the SAIL library file. Cards between these directives and the next Change directive on the SAIL input file will be inserted after sequence number N1. Individual columns of any sequenced card image can be altered by:

```
*C N1(C1,C2,NC1,NC2).
```

This directive will replace columns C1 through C2 on the SAIL library file with columns NC1 through NC2 from the card immediately following this change directive on the SAIL input file. Card images can be moved from one part of the SAIL library file to another by the directive:

```
*M N1,N2.
```

This directive must be preceded by one of the Change directives *A, *I, *C, or *D to define the sequence number for insertion of the card images between sequence numbers N1 and N2. The appearance of an *M directive does not terminate the definition of an insertion segment on the SAIL input file. *M directives can be mixed with other data for insertion or with other *M directives. The sequence numbers N1 through N2 are not altered in their original location.

2. SAIL Maintenance Directives

Even though SAIL library file maintenance has been relegated to the position of tail-end Charlie in this section, it is of importance because it keeps the system running and can simplify the use of the constructs. It was the intention of the HULL code authors to internally document the code rather than rely on external means of description. The good initial intentions were subverted by the volatile evolution of both HULL and SAIL. Both codes evolved to meet user needs rather than from a previously defined master plan. To meet the needs of day-to-day use, and to

give some appearance of orderly growth to the library file, the Maintenance directives in Table 4 were devised. The directives preceded by a C in Table 4 are also Change directives. Those preceded by an E are also Executive directives. Both types have been previously described. SAIL Executive assembly does not produce permanent changes to the SAIL library file. When the library file is updated to produce a new version, the Maintenance directives are instrumental in either implementing the differences between the old and new versions, or in facilitating the use of the library file listing for making such changes.

The *P directive, like the *B directive, forces a SAIL list run to the top of the next page and places the subroutine name field in the banner at the top of that page and all following pages until a subsequent *P or *B directive is encountered. The *P directive does not affect processing in any way.

TABLE 4. SAIL MAINTENANCE DIRECTIVES

	VERB	NOUN
C	*A	N1
E	*B	PROGRAM NAME
C	*C	N1,N2
C	*C	N1,(C1,C2,NC1,NC2)
C	*D	N1,N2
	*DIR	
E	*E	
	*EDIR	
C	*M	N1,N2
	*P	SUBROUTINE NAME

These directives are most commonly used to add new programs and subroutines to an existing SAIL library file. The *B directive would be used to define the start of each new program entity and the *P directive would be used to define the start of all subsequent subroutines in the inserted program. The *B directive is required to define the start of the new program for Executive assembly processing.

The *DIR and *EDIR directives are an aid to internal documentation. Sequenced card-images following a *DIR and preceding a *EDIR directive are listed during the directory list run by SAIL of the library file. In this way, collections of FORTRAN and SAIL comments cards can be assembled in a single listing for

portions or all of a SAIL library file. *P, *DIR, and *EDIR directives cause no change in the Executive assembly processing flow.

Comments which are to appear only in the SAIL listing can be inserted by putting an equal (=) symbol in column one of the sequenced card-image.

SECTION IV
SAIL INPUT/OUTPUT

The structure of the SAIL library file was previously defined in SECTION II. The physical library file serves as one element of the input data to SAIL during all modes of operation except an initial generate run. The SAIL library file has the logical name OLD when it is used as input data. When a library file is being produced by a Generate, Copy, or Update run, the new library will be put on logical file NEW. During an Executive assembly run or a Punch run, SAIL places its processed card-images on logical file SAIL. The compiler must therefore be instructed that its input data is on logical file SAIL. The normal OUTPUT file is used to produce the results of SAIL library list runs and messages concerning the execution of SAIL. Additional non-fatal error messages are output on file SSSSER during Update runs.

Primary control of SAIL is through the INPUT data stream. This is the file normally placed immediately after the control card data stream on most operating systems. On CDC systems SAIL searches the input stream for a record that begins with the word SAIL. If found, the data on INPUT will be used to select the mode of the SAIL run and provide change directives or other data to SAIL to complete the run.

The third input file used by SAIL is INPUT2. It is only used during Executive assembly runs. INPUT2 supplements (and overrides) data on the first two records of the library file (i.e. default program names and option definitions) and the primary INPUT file. When SAIL is used to support HULL or EPIC for instance, a pre-processor runs before SAIL execution to assemble INPUT2 from restart tapes or hydro-code problem input decks. A summary of these files is tabulated below:

INPUT/OUTPUT SAIL FILES

LOGICAL NAME	PURPOSE	INPUT(I) OUTPUT(O)	CDC NAME	IBM NAME	HONEYWELL NAME
OLD	Old Library	I	OLD	FT02F001	2
NEW	New Library	O	NEW	FT01F001	1
SAIL	Processed Card-images	O	SAIL	FT08F00N	15,16,..
INPUT	Control and Changes	I	INPUT	FT05F001	I*
INPUT2	Alternate Control	I	INPUT2	FT09F004	9
OUTPUT	Lists and Messages	O	OUTPUT	FT06F001	*
ERROR	Non-fatal Errors	O	SSSSER	FT04F001	4

1. MODE Parameters

The SAIL code can be viewed as seven different program entities which have only one thing in common; they all either use the SAIL library as input, output, or both. The selection of these different modes of operation is done by insertion of a Mode Parameter in the SAIL INPUT file. The following table shows the relationships between various files and operational modes of SAIL.

SAIL MODE PARAMETERS AND FILE FUNCTIONS

MODE	INPUT FILES	OUTPUT FILES	FUNCTION	RESULT
EXECUTIVE	Input,Input2, Old	SAIL	Produce compiler Input	Compiler input file
UPDATE	Input, Old	NEW	Produce New Library File	Updated or corrected Library
LIST	Input, Old	OUTPUT	List library file and attributes	Complete or partial listing
COPY	Input, Old	NEW	Prepare library file for transport	Library in different form
GENERATE	Input	NEW	Initiate new system	Library File
SCAN	Input, Old	OUTPUT	Locate Text	Text & sequence numbers of referenced text
PUNCH	Input, Old	SAIL	Reproduce library file in 80 column records	Card-Image Copy of library

A. EXECUTIVE MODE

The Executive mode is defaulted. The absence of a mode parameter or the existence of even a SAIL INPUT file will cause SAIL to begin Executive assembly of the default programs with the aid of the existing option definitions. This is the easiest and least flexible means of using SAIL. If a SAIL INPUT file is

present, additional parameters can be used to modify the SAIL Executive functions and thereby change the compileable output file. These parameters must be found between the initial keyword SAIL and the first input card or record which begins with an asterisk (a Change or Executive directive). The parameters which may be invoked during Executive assembly are listed in the following table:

TABLE 5. EXECUTIVE MODE INPUT PARAMETERS

VERB	VALUE	FUNCTION
DELOPTIONS	Option name(s)	delete options during this run
LINENO		insert SAIL sequence numbers in columns 73-80 of compileable file
OPTIONS ENDOPTIONS	Option name-value pairs	Add new option names and their values for this run
PROGRAM ENDPROGRAM	Program names	indicate programs to be processed for this run
PROSNAME	Program name	indicate special processing of a program

The verb or input parameter name of most of these can be used in other SAIL modes but with differing results. Care should be taken to note these distinctions.

The LINENO parameter is self explanatory. It allows the user to establish a one-to-one correspondence between a compilation listing and a SAIL library list. This simplifies program development, debug, and modification.

The DELOPTIONS and OPTIONS/ENDOPTIONS parameters are used to remove, change or add options to the option directory for this Executive assembly run. A file for which option A=2 and B=3 when processed by SAIL with the input card:

```
SAIL DELOPTIONS A OPTIONS B=2 C=4 ENDOPTIONS
```

would result in a new option directory with entries B=2 and C=4. The original options on file OLD remain unchanged. The OPTIONS parameter can be used to establish an option table. The parameter "AFTER" (including quotes) is used in the option definition

parameter list. The option immediately following "AFTER" is the option table name. Option name/value pairs following the table name will become members of the table. THUS:

```
SAIL OPTIONS TAB=2 "AFTER" TAB ENZONE=3
ENTWO=2 ENDOPTIONS
```

will produce the option table sequence TAB=2, ENZONE=3, and ENTWO=2. This table will exist only during this Executive assembly run.

The input parameters PROGRAM and PROSNAME are used to override the SAIL library file default program definitions. Program names following the input parameter PROGRAM will be assembled by SAIL. The PROSNAME parameter is used to designate the single program to be assembled for compilation. Other PROGRAM parameter defined names will be scanned by SAIL for option and procedure definitions but those programs will not appear in their entirety as compilable output on file SAIL. The PROGRAM, PROSNAME and OPTIONS input parameters are also used in establishing SAIL Executive assembly through the INPUT2 file.

B. UPDATE MODE

The second most used mode of SAIL is UPDATE. The following table lists the input parameters which may be used during UPDATE runs.

TABLE 6. UPDATE MODE INPUT PARAMETERS

VERB	VALUE	FUNCTION
EDIT	Character strings	String replacement
LINES	Number	define number of lines/page
SEQ		resequence file NEW
NOLIST		suppresses listing of NEW
OPTIONS ENDOPTIONS	Option name	change options on file NEW
PROGRAM	Program name	change default programs on NEW
SEQ PROGRAM	Program name	resequence listed programs

The UPDATE mode is invoked by the Keyword UPDATE to produce a new library file NEW with modifications determined by input parameters and Change directives.

The use of the parameters SEQ or SEQPROGRAM will resequence the entirety of file new or those specified by the SEQPROGRAM list respectively. A specification of Lines=60 or Lines=80 will define the two most commonly used print line intervals for the output printed on file OUTPUT. The NOLIST parameter inhibits the production of a library listing.

SAIL does not process directives during UPDATE runs. The result of the UPDATE run is a modified default program list specified by the PROGRAM input parameter, a modified option directory as defined by the OPTION input parameter and changes to the body of the library file as defined by the change directives on file INPUT.

C. LIST MODE

The LIST mode is used to list all or selected portions of file OLD on the output file. The contents of the first two header records are printed first. The last segment of information printed on a list run is a summary index which lists the sequence number of each program or subroutine as defined by *B or *P directives respectively. A LIST will always be produced during an UPDATE run unless the keyword NOLIST is inserted. During any list run, each program or subroutine unit will start at the top of a page with a header line identifying both program and subroutine names. Each line or card image will be preceded by its sequence number. Lines are terminated by the date of initial program generation or the lastest change to that sequence number. The line sequence number will be preceded by an asterisk if the last update of the file resulted in a change to that line of code. The asterisk will be eliminated if the keyword NOAST is included in the input data stream. The keyword LINES followed by an integer less than 85 will establish the number of sequence lines to be printed on each page. The normal value is 60 for 6 lines/inch and 80 for 8 lines/inch. The default value is 60.

To obtain a listing of the entire file, the input should be SAIL LIST. If it is desired to list only selected programs then the input syntax is:

```
SAIL LIST PROGRAM P1 P2 ... PN
```

where P1, P2, etc. are program names that are to be listed during this list run. If only the program documenting information

contained between *DIR/*EDIR pairs is desired, then the input will be:

SAIL LIST DIRECTORY

and all directory information for all programs on the file will be listed. If only the directory for selected programs are desired, then

SAIL LIST DIRECTORY PROGRAM P1, P2 ... PN

will produce directory listings of programs P1, P2, etc. If it is desirable to list full programs and selected program directories during the same list run, then the input would be:

SAIL LIST PROGRAM P1 "DIR" P2.

Program name P2, would be listed in its entirety while only the directory of program P1 would be listed since its name is followed by the modifier "DIR".

All of the LIST features are included to facilitate file maintenance. By including the change directives after the SAIL LIST input, the listing will incorporate changes, additions and deletions with the sequence number replaced by the word NEW. This is a convenient way of checking the effects of change directives on the file before doing a file UPDATE.

D. COPY MODE

The COPY mode of operation is similar to UPDATE except change directives are not honored. File OLD is copied to file NEW with changes in default programs and options as indicated by the PROGRAM and OPTIONS input parameters respectively. The CONVERT input parameter can be used to change the file format. It is order dependent. THUS:

SAIL COPY CONVERT

will convert a packed internal representation file OLD to a coded NEW,

SAIL CONVERT COPY

converts a coded file OLD into a packed representation on NEW. This is especially valuable for transporting a file from one operating system to another.

E. SCAN MODE

In this mode of operation a search is made for strings of characters which appear between the input parameters SCAN and ENDSCAN on the SAIL INPUT file. The strings to be found are delimited by any arbitrary character. Each string to be found is contained within a separate delimiter pair. The sequence number and card-images of all SAIL file records which contain the string are listed on file output. THUS:

```
SAIL SCAN
@ IF(J.EQ.1)@
@ DO 10@
ENDSCAN
```

will result in location and output of the sequenced card-images on the library file which contain the strings IF (J.EQ.1) or DO 10. If it is desired to substitute a string for an existing character string, the EDIT input parameter can be used in the UPDATE mode by following the word UPDATE by EDIT, listing the string to be replaced and the new string separately delimited, one pair to a record, and then terminating the input by ENDEDIT. For example:

```
SAIL UPDATE EDIT
@ DO 10@   @DO 20@
@ IF(J.EQ.1)@   @IF(K.EQ.1)@
ENDEDIT
```

would result in the left string on each card being located on file OLD and the right string substituted with the change on file NEW. The effects of such string substitutions can be examined by replacing UPDATE by LIST and producing a listing which can be visually checked. Both SCAN and EDIT can be modified by PROGRAM parameter definitions.

F. PUNCH MODE

If it is desirable to produce an 80 card image file of the SAIL library without sequence numbers and in single card records, then the PUNCH MODE can be used. The results are placed on file SAIL. All Executive directives originally in file OLD will be reproduced on SAIL. Specific programs can be selected by the PROGRAM parameter.

G. GENERATE MODE

As the name implies this mode is used to establish a new SAIL library file. The parameters OPTIONS and PROGRAM will establish the default values of their respective values. In addition the file name must be defined. This is done by the parameter SYSTEM=Name. The last input parameter must be GENERATE. This keyword is followed by the data to be placed on the new SAIL library - file NEW. If the first input cards following GENERATE are not preceded by the *B directive, this set will constitute the PROLOGUE. The input file should not contain imbedded End-Of-Files. The *E directive can be inserted to serve this function.

2. FILE PARAMETERS

The last set of input parameters is for checking or establishing the form of files OLD and NEW. One of these parameters; SYSTEM has been previously discussed in conjunction with the GENERATE parameter. In general, on CDC machines, SAIL expects file OLD to be previously defined as local by either an attach or tape request. If OLD does not exist, SAIL will search for a file as defined in a macro entry. If OLD exists on tape, then a tape request internal to SAIL can be made by the input parameter:

```
TAPE VSN = NNNNDD(NNN-DD)
```

where NNNN or NNN is the tape VSN and DD is the tape density (HY, NT, PE) and the character (-) is needed to fill the field out to six characters. This definition is positionally dependent. For instance in UPDATE Mode the SAIL input:

```
SAIL TAPE VSN=ABB-HY UPDATE
TAPE VSN = AAA-PE
```

implies file OLD is on tape ABB and file NEW will be placed on AAA. If only file OLD is involved in the particular SAIL Mode (Executive, List, Scan and Punch) then the order is unimportant since the intent is unambiguous. For a GENERATE run only file NEW is used and again the meaning is clear.

The input parameter SYSTEM can be used to assure that the correct file is being processed as well as establishing a new system name during COPY and UPDATE runs. Again the position before COPY or UPDATE is a reference to file OLD. If the input parameter system name does not match that on file OLD, SAIL will abnormally terminate. If the SYSTEM parameter is placed after the keywords COPY or UPDATE, file NEW will have the name indi-

cated, otherwise it will be copied from OLD. During UPDATE runs, the version number is normally incremented by one. The parameter VERSION can be used before and after the word UPDATE or COPY with results similar to those described for SYSTEM, except of course the version number will be affected.

An example of a set of inputs illustrating these are:

```
SAIL SYSTEM EPIC
OPTIONS COMPUTER=1  HULLSIZE=1000  ENDOPTIONS GENERATE
d1
d2
.
dn
```

which generates a new SAIL library file named EPIC with default options COMPUTER=1 and HULLSIZE=1000. Version number will default to one. All programs defined by *B directives will be placed in the default list.

```
SAIL SYSTEM EPIC VERSION 1 OPTIONS COMPUTER=3
ENDOPTIONS
```

will result in Executive assembly of all default programs with the option COMPUTER=3. The input parameters SYSTEM EPIC VERSION 1 were used only for assurance that the correct file was processed.

```
SAIL VERSION 1 UPDATE SYSTEM EPIC
PROGRAM PREP OPTIONS COMPUTER=4 ENDOPTIONS
```

will produce a new Version=2 EPIC library on file NEW. The entire default program list will be replaced by program PREP and the option directory will be modified to reflect the change in the value of option COMPUTER.

SECTION V
RUNNING UNDER SAIL

Since SAIL is a rather large code for compilation before each execution, most installations place it in a user library in the form of an absolute load. Since SAIL itself is maintained by the SAIL system, once a running absolute is in being, additional system changes for transport to other facilities are in hand.

If a particular batch job is to invoke SAIL several times in the same JOB submission, and the user does not wish the same data to be used, special precautions must be taken. SAIL, when looking for input, will go no further than the first occurrence of the word SAIL in the input stream. To circumvent this difficulty, the different segments of input pertinent to each execution of SAIL can be copied to alternate files; say A, B, and C. Sequential execution can then be done by using the input parameter I on the SAIL card:

```
SAIL(I=A)
SAIL(I=B)
SAIL(I=C)
```

This in an important consideration for making HULL run with SAIL or producing EPIC runs with SAIL since both of these codes use preprocessors for constructing alternate INPUT2 files.

A more exhaustive treatise on the automatic attachment of files and use of SAIL on other machines is contained in Reference 3. Good Sailing.

REFERENCES

1. Matuska, D.A. and R.E. Durrett, Version 1, The HULL Code, Air Force Weapons Laboratory, Kirtland AFB, NM, October, 1971.

2. Matuska, D.A. and L.P. Gaby II, THE SAIL UPDATE AND EXECUTIVE PROGRAM, AFWL/DYT-TN-75-3, Air Force Weapons Laboratory, Kirtland AFB, NM, May, 1975.

3. Gaby, L.P. II, D.C. Graham, and C.E. Rhoades, SAIL, AN AUTOMATED APPROACH TO SOFTWARE MANAGEMENT, AFWL-TR-78-80, Air Force Weapons Laboratory, Kirtland AFB, NM, January, 1979.

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Technical Info Center ATTN: DDC-DDA Cameron Station Alexandria, VA 22314
1	Director Defense Advanced Research Projects Agency ATTN: Tech Info 1400 Wilson Boulevard Arlington, VA 22209
1	Director Defense Nuclear Agency Washington, DC 20305
1	Deputy Assistant Secretary of the Army (R&D) Department of the Army Washington, DC 20310
2	Commander US Army BMD Advanced Technology Center ATTN: BMDATC-M, Mr. P. Boyd Mr. S. Brockway PO Box 1500 Huntsville, AL 35807
1	HQDA (DAMA-ARP) WASH DC 20310
1	HQDA (DAMA-MS) WASH DC 20310
2	Commander US Army Engineer Waterways Experiment Station ATTN: Dr. P. Hadala Dr. B. Rohani PO Box 631 Vicksburg, MS 39180
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMD-ST 5001 Eisenhower Avenue Alexandria, VA 22333
10	Commander US Army armament Research and Development Command ATTN: DRDAR-TD, Dr. R. Weigle DRDAR-LC, Dr. J. Frasier DRDAR-SC, Dr. D. Gyorog DRDAR-LCF, G. Demitrack DRDAR-LCA, G. Randers-Pehrson DRDAR-SCS-M, R. Kwatnoski DRDAR-LCU, E. Barrieres DRDAR-SCM, Dr. E. Bloore DRDAR-TSS (2 cys) Dover, NJ 07801
2	Director US Army ARRADCOM Benet Weapons Laboratory ATTN: DRDAR-LCB-TL Dr. Joseph E. Flaherty Watervliet, NY 12189
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61299
1	Commander US Army Aviation Research and Development Command ATTN: DRDAV-E 4300 Goodfellow Boulevard St. Louis, MO 63120

DISTRIBUTION LIST

No. of Copies	Organization
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Communications Research and Development Command ATTN: DRDCO-PPA-SA Fort Monmouth, NJ 07703
1	Commander US Army Electronics Research and Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703
3	Commander US Army Missile Research and Development Command ATTN: DRSMI-R DRSMI-RBL DRSMI-YDL Redstone Arsenal, AL 35809
2	Commander US Army Tank-Automotive Research and Development Command ATTN: DRDTA-UL V. H. Pagano Warren, MI 48090
1	Commander TARADCOM Tank-Automotive Systems Laboratory ATTN: T. Dean Warren, MI 48090
6	Director US Army Materials and Mechanics Research Center ATTN: DRXMR-T, Mr. J. Bluhm Mr. J. Mescall Dr. M. Lenoe R. Shea F. Quigley DRXMR-ATL Watertown, MA 02172
2	Commander US Army Research Office ATTN: Dr. E. Saibel Dr. G. Mayer PO Box 12211 Research Triangle Park NC 27709
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL (Tech Lib) Whie Sands Missile Range NM 88002
1	Office of Naval Research Department of the Navy ATTN: Code ONR 439, N. Perrone 800 North Quincy Street Arlington, VA 22217
3	Commander Naval Air Systems Command ATTN: AIR-604 Washington, DC 20360

DISTRIBUTION LIST

No. of Copies	Organization
2	Commander Naval Air Development Center, Johnsville Warminster, PA 18974
1	Commander Naval Missile Center Point Mugu, CA 93041
2	Naval Ship Engineering Center ATTN: J. Schell Tech Lib Washington, DC 20362
1	Commander & Director David W. Taylor Naval Ship Research & Development Center ATTN: Code 1740.4, R.A. Gramm Bethesda, MD 20084
3	Commander Naval Surface Weapons Center ATTN: Dr. W. G. Soper Mr. N. Rupert Code G35, D.C. Peterson Dahlgren, VA 22448
10	Commander Naval Surface Weapons Center ATTN: Dr. S. Fishman (2 cys) Code R-13, F.J. Zerilli K. Kim E.T. Toton M.J. Frankel Code U-11, J.R. Renzi R.S. Gross Code K-22, F. Stecher J.M. Etheridge Silver Spring, MD 20084
3	Commander Naval Weapons Center ATTN: Code 31804, Mr. M. Smith Code 326, Mr. P. Cordle Code 3261, Mr. T. Zulkoski China Lake, CA 93555
6	Commander Naval Weapons Center ATTN: Code 3181, John Morrow Code 3261, Mr. C. Johnson Code 3171, Mr. B. Galloway Code 3831, Mr. M. Backman Mr. R.E. VanDevender,Jr. Dr. O.E.R. Heimdahl China Lake, CA 93555
2	Director Naval Research Laboratory ATTN: Dr. C. Sanday Dr. H. Pusey Washington, DC 20375
2	Superintendent Naval Postgraduate School ATTN: Dir of Lib Dr. R. Ball Monterey, CA 93940
3	Long Beach Naval Shipyard ATTN: R. Kessler T. Eto R. Fernandez Long Beach, CA 90822
1	HQ USAF/SAMI Washington, DC 20330
1	AFIS/INOT Washington, DC 20330
20	ADTC/DLJW (MAJ G. Spitale) Eglin AFB, FL 32542
10	ADTC/DLYV (Mr. J. Collins) Eglin AFB, FL 32542
1	AFATL/DLYV Eglin AFB, FL 32542
1	AFATL/DLODL Eglin AFB, FL 32542

DISTRIBUTION LIST

No. of Copies	Organization
1	AFATL/CC Eglin AFB, FL 32542
1	AFATL/DLODR Eglin AFB, FL 32542
1	HQ PACAF/DOOQ Hickam AFB, HI 96853
1	HQ PACAF/OA Hickam AFB, HI 96853
1	OOALC/MMWMC Hill AFB, UT 84406
1	HQ TAC/DRA Langley AFB, VA 23665
1	TAC/INAT Langley AFB, VA 23665
1	AUL-LSE 71-249 Maxwell AFB, AL 36112
1	AFWAL/MLLN (Mr. T. Nicholas) Wright-Patterson AFB, OH 45433
1	ASD/ENESS (S. Johns) Wright-Patterson AFB, OH 45433
1	ASD/ENFEA Wright-Patterson AFB, OH 45433
1	ASD/XRP Wright-Patterson AFB, OH 45433
1	HQUSAFE/DOQ APO New York 09012
1	COMIPAC/I-32 Box 38 Camp H. I. Smith, HI 96861
10	Battelle Northwest Laboratories PO Box 999 ATTN: G. D. Marr Richland, WA 99352
4	Lawrence Livermore Laboratory PO Box 808 ATTN: Dr. R. Werne Dr. J.O. Hallquist Dr. M. L. Wilkins Dr. G. Goudreau Livermore, CA 94550
6	Los Alamos Scientific Laboratory PO Box 1663 ATTN: Dr. R. Karpp Dr. J. Dienes Dr. J. Taylor Dr. E. Fugelso Dr. D. E. Upham Dr. R. Keyser Los Alamos, NM 87545
6	Sandia Laboratories ATTN: Dr. R. Woodfin Dr. M. Sears Dr. W. Herrmann Dr. L. Bertholf Dr. A. Chabai Dr. C. B. Selleck Albuquerque, NM 87115
1	Headquarters National Aeronautics and Space Administration Washington, DC 20546
1	Jet Propulsion Laboratory 4800 Oak Grove Drive ATTN: Dr. Ralph Chen Pasadena, CA 91102
1	Director National Aeronautics and Space Administration Langley Research Center Langley Station Hampton, VA 23365

DISTRIBUTION LIST

No. of Copies	Organization
1	US Geological Survey 2255 N. Gemini Drive ATTN: Dr. D. Roddy Flagstaff, AZ 86001
1	AAI Corporation PO Box 6767 ATTN: R. L. Kachinski Baltimore, MD 21204
1	Aerojet Ordnance Company 9236 East Hall Road Downey, CA 90241
1	Aeronautical Research Associates of Princeton, Inc. 50 Washington Road Princeton, NJ 08540
1	Aerospace Corporation 2350 E. El Segundo Blvd. ATTN: Mr. L. Rubin El Segundo, CA 90009
1	AVCO Systems Division 201 Lowell Street ATTN: Dr. Reinecke Wilmington, MA 01803
4	Battelle Columbus Laboratories 505 King Avenue ATTN: Dr. M. F. Kanninen Dr. G. T. Hahn Dr. L. E. Hulbert Dr. S. Sampath Columbus, OH 43201
4	Boeing Aerospace Company ATTN: Mr. R. G. Blaisdell (M.S. 40-25) Dr. N. A. Armstrong, C. J. Artura (M.S. 8C-23) Dr. B. J. Henderson (M.S. 43-12) Seattle, WA 98124
2	Brunswick Corporation 4300 Industrial Avenue ATTN: P. S. Chang R. Grover Lincoln, NE 68504
1	Computer Code Consultants, Inc. 1680 Camino Redondo ATTN: Dr. Wally Johnson Los Alamos, NM 87544
1	Dresser Center PO Box 1407 ATTN: Dr. M.S. Chawla Houston, TX 77001
1	Effects Technology, Inc. 5383 Hollister Avenue Santa Barbara, CA 93111
1	Electric Power Research Institute PO Box 10412 ATTN: Dr. George Sliter Palo Alto, CA 94303
2	Firestone Defense Research and Products 1200 Firestone Parkway ATTN: R. L. Woodall L. E. Vescelius Akron, OH 44317
1	FMC Corporation Ordnance Engineering Division San Jose, CA 95114
1	Ford Aerospace and Communications Corporation Ford Road, PO Box A ATTN: L. K. Goodwin Newport Beach, CA 92660
1	General Atomic Company PO Box 81608 ATTN: R. M. Sullivan F. H. Ho S. Kwei San Diego, CA 92138

DISTRIBUTION LIST

No. of Copies	Organization
1	General Dynamics PO Box 2507 ATTN: J. H. Cuadros Pomona, CA 91745
1	General Electric Company Lakeside Avenue ATTN: D. A. Graham, Room 1311 Burlington, VT 05401
1	President General Research Corporation ATTN: Lib McLean, VA 22101
1	Goodyear Aerospace Corporation 1210 Massilon Road Akron, OH 44315
1	H. P. White Laboratory 3114 Scarboro Road Street, MD 21154
5	Honeywell, Inc. Government and Aerospace Products Division ATTN: Mr. J. Blackburn Dr. G. Johnson Mr. R. Simpson Mr. K. H. Doeringsfeld Dr. D. Vavrick 600 Second Street, NE Hopkins, MN 55343
1	Hughes Aircraft Corporation ATTN: Mr. W. Keppel MS M-5, Bldg. 808 Tucson, AZ 85706
2	Kaman Sciences Corporation 1500 Garden of the Gods Road ATTN: Dr. P. Snow Dr. D. Williams Colorado Springs, CO 80933
1	Lockheed Palo Alto Research Laboratory 3251 Hanover Street ATTN: Org 5230, Bldg. 201 Mr. R. Robertson Palo Alto, CA 94394
1	Lockheed Missiles and Space Company PO Box 504 ATTN: R. L. Williams Dept. 81-11, Bldg. 154 Sunnyvale, CA 94086
1	Materials Research Laboratory, Inc. 1 Science Road Glenwood, IL 60427
2	McDonnell-Douglas Astronautics Company 5301 Bolsa Avenue ATTN: Dr. L. B. Greszczuk Dr. J. Wall Huntington Beach, CA 92647
1	New Mexico Institute of Mining and Technology ATTN: TERA Group Socorro, NM 87801
1	Northrup Corporation 3901 W. Broadway ATTN: R. L. Ramkumar Hawthorne, CA 90250
1	Nuclear Assurance Corporation 24 Executive Park West ATTN: T. C. Thompson Atlanta, GA 30245
2	Orlando Technology, Inc. PO Box 855 ATTN: Mr. J. Osborn Mr. D. Matuska Shalimar, FL 32579

DISTRIBUTION LIST

No. of Copies	Organization
1	Pacific Technical Corporation 460 Ward Drive ATTN: Dr. F. K. Feldmann Santa Barbara, CA 93105
1	Rockwell International Missile Systems Division ATTN: A. R. Glaser 4300 E. Fifth Avenue Columbus, OH 43216
3	Schumberger Well Services Perforating Center ATTN: J. E. Brooks J. Brookman Dr. C. Aseltine PO Box A Rosharon, TX 77543
1	Science Applications, Inc. 101 Continental Boulevard Suite 310 El Segundo, CA 90245
1	Ship Systems, Inc. 11750 Sorrento Valley Road ATTN: Dr. G. G. Erickson San Diego, CA 92121
1	Systems, Science and Software PO Box 1620 ATTN: Dr. R. Sedgwick La Jolla, CA 92038
2	TRW One Space Park, R1/2120 ATTN: D. Ausherman M. Bronstein Redondo Beach, CA 90277
1	United Technologies Research Center 438 Weir Street ATTN: P. R. Fitzpatrick Glastonbury, CT 06033
1	US Steel Corporation Research Center 125 Jamison Lane Monroeville, PA 15146
1	VPI & SU 106C Norris Hall ATTN: Dr. M. P. Kamat Blacksburg, VA 24061
2	Vought Corporation PO Box 225907 ATTN: Dr. G. Hough Dr. Paul M. Kenner Dallas, TX 75265
1	Westinghouse, Inc. PO Box 79 ATTN: J. Y. Fan W. Mifflin, PA 15122
1	Drexel University Department of Mechanical Engr. ATTN: Dr. P. C. Chou 32d and Chestnut Streets Philadelphia, PA 19104
3	Southwest Research Institute Dept. of Mechanical Sciences ATTN: Dr. U. Lindholm Dr. W. Baker Dr. R. White 8500 Culebra Road San Antonio, TX 78228
4	SRI International 333 Ravenswood Avenue ATTN: Dr. L. Seaman Dr. L. Curran Dr. D. Shockey Dr. A. L. Florence Menlo Park, CA 94025
2	University of Arizona Civil Engineering Department ATTN: Dr. D. A. DaDeppo Dr. R. Richard Tucson, AZ 85721

DISTRIBUTION LIST

No. of Copies	Organization
1	University of Arizona School of Engineering ATTN: Dean R. Gallagher Tucson, AZ 85721
1	University of California Los Angeles ATTN: Dr. M. Ziv Los Angeles, CA 90024
1	University of California Department of Physics ATTN: Dr. Harold Lewis Santa Barbara, CA 93106
2	University of California College of Engineering ATTN: Prof. W. Goldsmith Dr. A. G. Evans Berkeley, CA 94720
2	University of Delaware Department of Mechanical Engineering ATTN: Prof. J. Vinson Prof. B. Pipes Newark, DE 19711
1	University of Denver Denver Research Institute ATTN: Mr. R. F. Recht 2390 S. University Blvd. Denver, CO 80210
2	University of Florida Department of Engineering Sciences ATTN: Dr. R. L. Sierakowski Dr. L. E. Malvern Gainesville, FL 32601
1	University of Oklahoma School of Aerospace, Mechanical and Nuclear Engineering ATTN: Dr. C. W. Bert Norman, OK 73019

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: DRXSY-D
DRXSY-MP, H. Cohen
Cdr, USATECOM
ATTN: DRSTE-TO-F
Dir, USA MTD
ATTN: Mr. S. Keithley

Dir, USACSL, EA
ATTN: DRDAR-CLB-PA
Bldg. E3516

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet, fold as indicated, staple or tape closed, and place in the mail. Your comments will provide us with information for improving future reports.

1. BRL Report Number _____

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

_____

_____

_____

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.) _____

_____

_____

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

_____

_____

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.) _____

_____

_____

_____

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: _____

Telephone Number: _____

Organization Address: _____

_____

_____