

LIST OF ERROR MESSAGES AND
DIAGNOSTICS FOR THE DAVE SYSTEM

by

Carol Miesse
and
Edward C. Read

Department of Computer Science
University of Colorado
Boulder, Colorado 80302

TR #CU-CS-074-75 July 1975

Supported in part by NSF Grant GJ-36461

This report is divided into 3 parts:

PART A is a list of specimen errors, warning and messages produced by the DAVE system in its action upon the programs and sub-programs being analyzed. (Pages 2-11)

PART B is a list of DAVE diagnostics which relate to the operation of the system itself. The explanation is brief and is directed to the DAVE user. (Pages 12-22)

PART C is a more complete list of DAVE diagnostics, oriented toward the DAVE maintainer. (Pages 23-44)

The DAVE system is under continuous development and the diagnostics listed here are those incorporated in DAVE 4.0 and are subject to change in future editions.

PART A

*****ERROR

A FUNCTION NAME IS INCORRECTLY USED FOR INPUT.

NAME OF FUNCTION	A
------------------	---

*****ERROR

STATEMENT NO.	1	BASIC BLOCK NO.	1
---------------	---	-----------------	---

AN ARGUMENT IN THE CALLING SUBPROGRAM IS AN EXPRESSION OR CONSTANT. THE CALLED SUBPROGRAM MAKES A VALUE ASSIGNMENT TO THE CORRESPONDING PARAMETER.

	CALLING SUBPROGRAM	CALLED SUBPROGRAM
	SYSMAIN	A
ARGUMENT POSITION	2	2
NAME OF ARGUMENT		D
KIND OF ARGUMENT	INTEGER CONSTANT	
INPUT CLASS		NON-INPUT
OUTPUT CLASS		STR. OUTPUT

*****ERROR

STATEMENT NO.	1	BASIC BLOCK NO.	1
---------------	---	-----------------	---

AN ILLEGAL SIDE EFFECT HAS BEEN DETECTED. IT OCCURS VIA A VARIABLE PASSED IN A PARAMETER LIST. THIS VARIABLE HAS APPEARED AT LEAST TWICE IN THIS STATEMENT -- IN ONE APPEARANCE IT IS USED AS STRICT INPUT AND IN THE OTHER AS STRICT OUTPUT.

	CALLING SUBPROGRAM	CALLED SUBPROGRAM
	SYSMAIN	A
ARGUMENT POSITION	2	2
NAME OF ARGUMENT	C	D

*****ERROR

STATEMENT NO.	1	BASIC BLOCK NO.	1
---------------	---	-----------------	---

THE NUMBERS OF ARGUMENTS IN THE PARAMETER LISTS DO NOT CORRESPOND.

	CALLING SUBPROGRAM	CALLED SUBPROGRAM
	SYSMAIN	A

*****ERROR

A COMMON VARIABLE IN THE CALLED SUBPROGRAM REQUIRES A VALUE, YET ITS COMMON BLOCK IS NOT IN THE CALLING SUBPROGRAM NOR IS ITS VALUE INITIALIZED IN A BLOCK DATA SUBPROGRAM,

	CALLING SUBPROGRAM	CALLED SUBPROGRAM
	SYSMAIN	A
COMMON BLOCK NAME		B
NAME OF VARIABLE		C

*****ERROR

SUB APPEARS TO BE A SUBPROGRAM THAT IS CALLED YET NOT AVAILABLE. EITHER AN ARRAY (WITH THE ABOVE NAME) IS NOT DIMENSIONED, OR THE SUBPROGRAM CODE IS MISSING.
NAME OF CALLER

A
B
C
D
E
F
G
H
I
J

*****ERROR

A COMMON VARIABLE REQUIRES A VALUE IN THE MAIN PROGRAM, YET IT IS NOT INITIALIZED IN BLOCK DATA.

CALLING SUBPROGRAM
SYSMAIN

COMMON BLOCK NAME B
VAR. NAME IN CALLER C
VAR. NAME, BLK. DATA D

*****ERROR

THE DUMMY ARGUMENT A IS ALSO IN COMMON AND IS ASSIGNED A VALUE, WHICH IS PROHIBITED.

*****WARNING

A FUNCTION NAME IS NOT ALWAYS ASSIGNED A VALUE.

NAME OF FUNCTION A

*****WARNING

STATEMENT NO. 1 BASIC BLOCK NO. 1
CORRESPONDING ARGUMENTS IN THE PARAMETER LISTS ARE OF DIFFERENT DATA TYPES.

CALLING SUBPROGRAM CALLED SUBPROGRAM
SYSMAIN A

ARGUMENT POSITION 2 REAL
DATA TYPE INTEGER
NAME OF ARGUMENT D
KIND OF ARGUMENT INTEGER CONSTANT
INPUT CLASS INPUT
OUTPUT CLASS NON-OUTPUT

*****ERROR STATEMENT NO. 1 BASIC BLOCK NO. 1
AN ARGUMENT IN THE CALLING PROGRAM IS A PROCEDURE DECLARED
EXTERNALLY. THE CORRESPONDING PARAMETER WILL BE USED BY THE
CALLED SUBPROGRAM TO SUPPLY A VALUE.

CALLING SUBPROGRAM CALLED SUBPROGRAM
SYSMAIN A
ARGUMENT POSITION 2 2
KIND OF ARGUMENT PROC. DEC. EXTERNAL
NAME OF PROCEDURE G
NAME OF ARGUMENT D
INPUT CLASS STR. INPUT
OUTPUT CLASS OUTPUT

*****ERROR STATEMENT NO. 1 BASIC BLOCK NO. 1
AN ARGUMENT IN THE CALLING PROGRAM IS A PROCEDURE DECLARED
EXTERNALLY. THE CORRESPONDING PARAMETER WILL RECEIVE A VALUE
IN THE CALLED SUBPROGRAM.

CALLING SUBPROGRAM CALLED SUBPROGRAM
SYSMAIN A
ARGUMENT POSITION 2 2
KIND OF ARGUMENT PROC. DEC. EXTERNAL
NAME OF PROCEDURE G
NAME OF ARGUMENT D
INPUT CLASS NON-INPUT
OUTPUT CLASS STR. OUTPUT

*****ERROR
THE VARIABLE NAMED A BECOMES UNDEFINED (FALLING THROUGH
STATEMENT NO. 1), YET IS USED AS INPUT ON ALL
PATHS THEREAFTER.
ONE SUCH PATH, INDICATED BY BASIC BLOCK NUMBERS, IS
1 2 3 4 5 6 7 8 9 10

*****ERROR
A FUNCTION NAME IS NEVER ASSIGNED A VALUE.

NAME OF FUNCTION A

*****ERROR
THE LOCAL VARIABLE NAMED A IS REFERENCED BEFORE BEING
DEFINED ON ALL PATHS.
ONE SUCH PATH, INDICATED BY BASIC BLOCK NUMBERS, IS
1 2 3 4 5 6 7 8 9 10

*****WARNING STATEMENT NO. 1 BASIC BLOCK NO. 1
CORRESPONDING ARGUMENTS IN THE PARAMETER LISTS ARE OF
DIFFERENT DIMENSIONALITY.

CALLING SUBPROGRAM CALLED SUBPROGRAM
SYSMAIN A
ARGUMENT POSITION 2 2
NUMBER OF DIMENSIONS 1 2
NAME OF ARGUMENT D D
KIND OF ARGUMENT IDENTIFIER
INPUT CLASS STR. INPUT
OUTPUT CLASS OUTPUT

*****WARNING STATEMENT NO. 1 BASIC BLOCK NO. 1
AN ARGUMENT IN THE CALLING SUBPROGRAM IS AN EXPRESSION OR
CONSTANT. THE CALLED SUBPROGRAM MAKES NO INPUT USE OF THE
CORRESPONDING PARAMETER.

CALLING SUBPROGRAM CALLED SUBPROGRAM
SYSMAIN A
ARGUMENT POSITION 2 2
NAME OF ARGUMENT D
KIND OF ARGUMENT INTEGER CONSTANT
INPUT CLASS
OUTPUT CLASS NON-INPUT
NON-OUTPUT

*****WARNING STATEMENT NO. 1 BASIC BLOCK NO. 1
A POSSIBLE ILLEGAL SIDE EFFECT HAS BEEN DETECTED.
IT OCCURS VIA A VARIABLE PASSED IN A PARAMETER LIST.
THIS VARIABLE HAS APPEARED AT LEAST TWICE IN THIS
STATEMENT -- IN ONE APPEARANCE IT IS USED AS INPUT
AND IN THE OTHER AS OUTPUT.

CALLING SUBPROGRAM CALLED SUBPROGRAM
SYSMAIN A
ARGUMENT POSITION 2 2
NAME OF ARGUMENT C D

*****WARNING
A VARIABLE IN A PARAMETER LIST IS USED FOR NEITHER INPUT NOR
OUTPUT.

NAME OF VARIABLE A

*****WARNING
A COMMON VARIABLE IN THE CALLED SUBPROGRAM MAY REQUIRE
A VALUE, YET ITS COMMON BLOCK IS NOT AVAILABLE TO THE
CALLING SUBPROGRAM AND THE VARIABLE IS NOT INITIALIZED
IN BLOCK DATA.

CALLING SUBPROGRAM CALLED SUBPROGRAM
SYSMAIN A
COMMON BLOCK NAME B
NAME OF VARIABLE C

*****WARNING

A COMMON VARIABLE IN THE CALLED SUBPROGRAM MAY BE ASSIGNED A VALUE, YET ITS COMMON BLOCK IS NOT AVAILABLE TO THE CALLING SUBPROGRAM. HENCE, A COMPUTED VALUE MAY BE LOST.

CALLING SUBPROGRAM CALLED SUBPROGRAM
SYSMAIN A

COMMON BLOCK NAME B
NAME OF VARIABLE C

*****WARNING

A COMMON VARIABLE IN THE CALLED SUBPROGRAM IS ASSIGNED A VALUE, YET ITS COMMON BLOCK IS NOT IN THE CALLING SUBPROGRAM. HENCE, A COMPUTED VALUE WILL BE LOST.

CALLING SUBPROGRAM CALLED SUBPROGRAM
SYSMAIN A

COMMON BLOCK NAME B
NAME OF VARIABLE C

*****WARNING

CORRESPONDING BLOCK COMMON VARIABLES HAVE DIFFERENT DATA TYPES IN THE CALLING AND CALLED SUBPROGRAMS.

CALLING SUBPROGRAM CALLED SUBPROGRAM
SYSMAIN A

COMMON BLOCK NAME B B
NAME OF ARGUMENT C E
DATA TYPE LOGICAL REAL

*****WARNING

STATEMENT NO. 1 BASIC BLOCK NO. 1
AN ARGUMENT IN THE CALLING PROGRAM IS A PROCEDURE DECLARED EXTERNALLY. THE CORRESPONDING PARAMETER MAY BE USED BY THE CALLED SUBPROGRAM TO SUPPLY A VALUE.

CALLING SUBPROGRAM CALLED SUBPROGRAM
SYSMAIN A

ARGUMENT POSITION 2 2
KIND OF ARGUMENT PROC. DEC. EXTERNAL
NAME OF PROCEDURE G
NAME OF ARGUMENT D
INPUT CLASS INPUT
OUTPUT CLASS OUTPUT

```

*****
*****WARNING          STATEMENT NO.    1  BASIC BLOCK NO.    1
AN ARGUMENT IN THE CALLING PROGRAM IS A PROCEDURE DECLARED
EXTERNALLY.  THE CORRESPONDING PARAMETER MAY RECEIVE A VALUE
IN THE CALLED SUBPROGRAM.

```

```

                CALLING SUBPROGRAM  CALLED SUBPROGRAM
                SYSMAIN              A
ARGUMENT POSITION          2          2
KIND OF ARGUMENT        PROC. DEC. EXTERNAL
NAME OF PROCEDURE              G
NAME OF ARGUMENT
INPUT CLASS                    D
OUTPUT CLASS                    STR. INPUT
                                OUTPUT
*****

```

```

*****WARNING
THE VARIABLE NAMED          A BECOMES UNDEFINED (FALLING THROUGH
STATEMENT NO.    1), YET IS USED AS INPUT ON SOME
PATHS THEREAFTER.
ONE SUCH PATH, INDICATED BY BASIC BLOCK NUMBERS, IS
1      2      3      4      5      6      7      8      9      10
*****

```

```

*****WARNING
THE VARIABLE NAMED          A IS ASSIGNED A VALUE
IN ITS LAST USAGE ON ALL PATHS.
ONE SUCH PATH, INDICATED BY BASIC BLOCK NUMBERS, IS
1      2      3      4      5      6      7      8      9      10
*****

```

```

*****WARNING
THE LOCAL VARIABLE NAMED          A IS REFERENCED BEFORE BEING
DEFINED ON SOME PATHS.
ONE SUCH PATH, INDICATED BY BASIC BLOCK NUMBERS, IS
1      2      3      4      5      6      7      8      9      10
*****

```

```

*****WARNING
A COMMON VARIABLE IN THE CALLED SUBPROGRAM REQUIRES A
VALUE, YET ITS COMMON BLOCK MAY NOT BE AVAILABLE TO THE
SUBPROGRAM AND THE VARIABLE IS NOT INITIALIZED IN A
BLOCK DATA SUBPROGRAM.

```

```

                CALLING SUBPROGRAM  CALLED SUBPROGRAM
                SYSMAIN              A
COMMON BLOCK NAME          B
NAME OF VARIABLE           C
*****

```

*****WARNING

A COMMON VARIABLE IN THE CALLED SUBPROGRAM IS ASSIGNED
A VALUE, YET ITS COMMON BLOCK MAY NOT BE AVAILABLE TO
THE CALLING SUBPROGRAM. HENCE, A COMPUTED VALUE MAY BE
LOST.

	CALLING SUBPROGRAM		CALLED SUBPROGRAM
		SYSMAIN	A

COMMON BLOCK NAME			B
-------------------	--	--	---

NAME OF VARIABLE			C
------------------	--	--	---

*****WARNING

A COMMON VARIABLE IN THE CALLED SUBPROGRAM, INITIALIZED
IN BLOCK DATA, MAY BE ASSIGNED A VALUE AND UNDEFINITION
MAY OCCUR UPON SUBSEQUENT INVOCATION OF THE CALLED
ROUTINE.

	CALLING SUBPROGRAM		CALLED SUBPROGRAM
		SYSMAIN	A

COMMON BLOCK NAME			B
-------------------	--	--	---

NAME OF VARIABLE			C
------------------	--	--	---

*****WARNING

A COMMON VARIABLE IN THE CALLED SUBPROGRAM, INITIALIZED
IN BLOCK DATA, IS ASSIGNED A VALUE AND UNDEFINITION
WILL OCCUR UPON SUBSEQUENT INVOCATION OF THE CALLED
ROUTINE.

	CALLING SUBPROGRAM		CALLED SUBPROGRAM
		SYSMAIN	A

COMMON BLOCK NAME			B
-------------------	--	--	---

NAME OF VARIABLE			C
------------------	--	--	---

*****WARNING

A COMMON VARIABLE MAY REQUIRE A VALUE IN THE MAIN
PROGRAM, YET IT IS NOT INITIALIZED IN BLOCK DATA.

	CALLING SUBPROGRAM	
		SYSMAIN

COMMON BLOCK NAME		B
-------------------	--	---

VAR. NAME IN CALLER		C
---------------------	--	---

VAR. NAME, BLK. DATA		D
----------------------	--	---

 *****WARNING

THE DUMMY ARGUMENT A IS ALSO IN COMMON AND MAY BE ASSIGNED
 A VALUE, WHICH IS PROHIBITED.

 *****WARNING

THE LOCAL VARIABLE NAMED A, INITIALIZED IN A DATA
 STATEMENT, IS USED AS OUTPUT AND UNDEFINITION MAY OCCUR
 UPON SUBSEQUENT INVOCATION OF THE SUBPROGRAM.

 *****WARNING

THE LOCAL VARIABLE NAMED A, INITIALIZED IN A DATA
 STATEMENT, IS USED AS STRICT OUTPUT AND UNDEFINITION
 WILL OCCUR UPON SUBSEQUENT INVOCATION OF THE SUBPROGRAM.

 *****MESSAGE

SUBPROGRAM A IS NEVER CALLED.

 *****MESSAGE

COMMON VARIABLE C IN BLOCK B OF
 SUBPROGRAM A IS INITIALIZED IN BLOCK DATA.

 *****MESSAGE

THE FOLLOWING COMMON BLOCKS, ALTHOUGH NOT EXPLICITLY IN
 SUBPROGRAM SYSMAIN ARE AVAILABLE TO IT.

COMMON BLOCK	AVAILABILITY
A	ALWAYS
C	SOMETIMES
E	ALWAYS
G	SOMETIMES
I	ALWAYS

 *****MESSAGE

STATEMENT NO. 1 BASIC BLOCK NO. 1
 THE FOLLOWING COMMON VARIABLES ARE USED AS INPUT WHEN
 SUBPROGRAM A IS CALLED.

CALLING SUBPROGRAM	CALLED SUBPROGRAM
SYSMAIN	A
COMMON BLOCK NAME	B
NAME OF VARIABLE	C
INPUT CLASS	STR. INPUT

```
*****  
****MESSAGE      STATEMENT NO.    1  BASIC BLOCK NO.    1  
  THE FOLLOWING COMMON VARIABLES ARE USED AS OUTPUT WHEN  
  SUBPROGRAM      A IS CALLED.  
                  CALLING SUBPROGRAM  CALLED SUBPROGRAM  
                  SYSMAIN              A  
  
COMMON BLOCK NAME      B  
NAME OF VARIABLE       C  
  OUTPUT CLASS         OUTPUT  
*****
```

PART B

	<u>NO. +</u>	<u>MESSAGE</u>	<u>MEANING</u>
* AC1	-1	*****	Check source for a syntax error in a logical or relational operator, or in a logical constant. [*****: system debugging information]
	-2	*****	Check source for a syntax error in a logical or relational operator, or in a logical constant. [*****: system debugging information]
AC2	-1	*****	System fault [*****: system debugging information]
AC3	-1	*****	A syntax error involving the use of * or ** has been encountered. [*****: system debugging information]
AC4	-1	*****	System fault [*****: system debugging information]
	-2	*****	A syntax error in a line of source containing .D or .E has been encountered. [*****: system debugging information]
AC5	-1	*****	A syntax error in a format specification involving E, D or H has been encountered. [*****: system debugging information]
	-3	*****	An illegal Hollerith specification has been encountered. [*****: system debugging information]

* Name of subroutine in which error occurred.

+ A positive error number indicates a non-fatal condition and execution continues; a negative error number indicates a fatal condition and execution terminates.

	<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
ADD	-1	KEY ERR	System fault
ASIGN	-1	IDX	System fault. Possible correction: the system places an upper limit of 20 on the number of parameters in an arithmetic statement function.
BLKTAB	-1	*****	System fault [*****: system debugging information]
CGRAPH	-1	ARRFULL	System fault. Possible correction: reduce the number of subprograms being processed simultaneously.
	-2	ARRFULL	System fault
CHR	-1	*****	An illegal FORTRAN character has been encountered. [***** gives the illegal character]
CMDOWN	-1	MISSING BLOCK	System fault
	-2	UNKNOWN AVAIL	System fault

	<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
COMTCH	-1	*****	System fault [*****: system debugging information]
	-2	*****	System fault [*****: system debugging information]
CONTOK	-1	BAD TYPE	System fault
	-2	NO END	System fault
	-3	BAD IN	System fault
	-4	BAD IN	System fault
	-5	BAD IN	System fault
DATYP	-1	KEY ERROR	System fault
ENPATH	-1	*****	System fault [*****: system debugging information]
EQCHN	-1	EQUIV ERR	System fault. Possible correction: the system places an upper limit of 100 on the number of variables which can be made (pairwise) equivalent.
	-2	EQUIV ERR	The same variable name has been incorrectly equivalenced twice.

	<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
EQUIV	-1	IQMAX	System fault. Possible correction: the system places an upper limit of 100 on the number of variables which can be made (pairwise) equivalent.
EXPRSN	-1	*****	System fault [*****: system debugging information]
	-2	*****	Check source for a syntax error in an expression: e.g., the illegal juxtaposition of two operators [*****: system debugging information]
	-3	*****	Check source for a syntax error in an expression: e.g., the illegal juxtaposition of two operators [*****: system debugging information]
FCALL	-1	STACKVALUEERROR	System fault
	-2	TOKEN ERROR	System fault
	-3	UNDERFLOW	System fault, or possible incorrect parentheses count in source code: occurring in a subprogram call or arithmetic statement function call.
FLTCHR	-1	*****	Syntax error in the expression of a real number, or system fault. [***** gives the invalid real number]

	<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
GRORDR	-1	BAD NAME	System fault
INPVAR	-1	UNDERFLOW	System fault
	-2	OVERFLOW	System fault
	-3	UNDEFINED INPUT	System fault
INTCHR	-1	*****	Syntax error in the expression of an integer or system fault. [***** gives the invalid integer representation]
IOINIT	-1	PASTEND	System fault
	-2	PASTEND	System fault. Possible correction: reduce the number of basic blocks, or the number of local or global variables.
	-3	PASTEND	System fault
	-4	UNCLASS	System fault
	-5	PASTEND	System fault
	-6	PASTEND	System fault
	-7	PASTEND	System fault
IOSTAT	-1	CODE ERR	System fault

	<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
IOVUND	-1	*****	System fault [*****: system debugging information]
	-2	*****	System fault [*****: system debugging information]
	-3	*****	System fault [*****: system debugging information]
ISTYPE	-1	*****	The use of an illegal character has been detected. [*****: system debugging information]
LABBLK	-1	*****	System fault [*****: system debugging information]
	-2	*****	System fault [*****: system debugging information]
	-3	*****	System fault [*****: system debugging information]
LASTNB	-1	*****	System fault [*****: system debugging information]
LEXST	1	*****	System fault [*****: system debugging information]
	2	*****	System fault [*****: system debugging information]
	3	*****	System fault [*****: system debugging information]

	<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
LUNTIL	-1	*****	A non-ANSI FORTRAN statement has been encountered. Check source code for missing parentheses in an IF statement or for a missing character (0) in an ASSIGN TO statement.
MKLBTB	-1	*****	There is a labelling error in an ASSIGN, GO TO, or DO statement. [*****: system debugging information]
NAMCHR	-1	*****	Syntax error in the expression of a real number, or system fault. [***** gives the invalid FORTRAN variable name representation]
NXTNB	-1	*****	System fault [*****: system debugging information]
OUTVAR	-1	UNDERFLOW	System fault
	-2	OVERFLOW	System fault
	-3	OVERFLOW	System fault
	-4	UNDEF	System fault

	<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
PHASE1	-1	*****	A syntax error in the source code has been detected. Check ASSIGN and DO statements. [*****: no message]
	-2	*****	A non-ANSI FORTRAN statement has been encountered [***** gives system debugging information]
PRACI		*****	System fault [*****: System debugging information]
PRODM	-1	NOTDM	System fault
	-2	NOLFTPAR	A syntax error in a dimension statement has been encountered. Check for a missing left parenthesis.
PUSH	-1	*****	System fault. [***** contains system debugging information]
PUTTOK	-1	*****	System fault [*****: system debugging information]
RDWR	-1	IM.GT.20	System fault. The system allows a maximum of 20 indices in a READ or WRITE statement.
	-3	NOT IDENT	Syntax error detected in a READ or WRITE statement.

	<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
SAVEIO	-2	VARDIM	An array in COMMON has been found to be variably dimensioned.
SBMTCH	-1	OVERFLOW	System fault
	-2	OVERFLOW	System fault
SKIP	-1	*****	A non-ANSI FORTRAN statement has been encountered. Check FORTRAN words with special status (e.g., ASSIGN, GO TO) for missing characters. [*****: no message]
STEPAR	-1	PAR ERR	System fault
STRDM	-1	BADTOK	There is an error in the syntax of a dimensioned variable appearing in a declaration statement.
		NORTPAR	A syntax error in a dimension statement has been encountered. Check for a missing right parenthesis.
TRANS	-1	*****	System fault [*****: system debugging information]
	-2	*****	System fault [*****: system debugging information]
	-3	*****	System fault [*****: system debugging information]
	-4	*****	System fault [*****: system debugging information]

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
VARPR	1 *****	An attempt has been made in line J of the source to supply an input value by means of the variable INPVAR at a point where INPVAR has no value assigned to it. [*****: supplies J, INPVAR]

PART C

AC1

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=KCHAR (2)=IPONT (3)=ISAVE	An operator or logical constant has not been found as expected. <u>Probable cause:</u> Syntax error
-2	(1)=INSTK(IPONT) (2)=IPONT (3)=IEND	A complete operator or logical constant was not found as expected. <u>Probable cause:</u> Syntax error.

AC2

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=INSTK(IPONT) (2)=IPONT (3)=ITYPE	The current character INSTK(IPONT) is not one of those expected. <u>Probable cause:</u> System fault.

AC3

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=IBEG (2)=IEND (3)=ISAVE	No non-blank character is found after the asterisk. <u>Probable cause:</u> A syntax problem involving the use of * or ** has been encountered.

AC4

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
01	ERRPKT(1)=IBEG (2)=IEND (3)=ISAVE	A non-blank character was not found as expected. <u>Probable cause:</u> System fault.
-2	ERRPKT(1)=IBEG (2)=IEND (3)=ISAVE	Either .D or .E was found. Alone they constitute an illegal string. <u>Probable cause:</u> Syntax error.

AC5

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=INST(IPONT) (2)=IPONT	The next character must be an E, D or H and it is not. <u>Probable cause:</u> Syntax error.
-3	ERRPKT(1)=INST(IPONT) (2)=IPONT	The length of the Hollerith string is less than or equal to 0. It must be a positive integer. <u>Probable cause:</u> Syntax error - an illegal Hollerith specification has been encountered.

ADD

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"KEY ERR"	Either KEY is less than 0 or KEY is greater than 4; KEY (which is required to classify the symbolic name indicated by INDX) is out of range. <u>Probable cause:</u> System fault.

ASIGN

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"IDX"	An overflow condition exists. IDUM is dimensioned IDUM(20); IDX, the index into IDUM, exceeds 20. <u>Probable cause:</u> System fault. The system places an upper limit of 20 on the number of parameters in an arithmetic statement function.

BLKTAB

-1	ERRPKT(1)=NUM ERRPKT(2)=LENGTH	NUM exceeds LENGTH(=400). The system places an upper limit of 400 on the number of blocks in the current sub-program. <u>Probable cause:</u> System fault.
----	-----------------------------------	---

CGRAPH

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-2	"ARRFULL"	An overflow condition in array *IORDR is indicated. IORDR has size IORSZ, and the pointer to the next position (INORDR) of array IORDR exceeds IORSZ. <u>Probable cause:</u> System fault. Reduce the number of programs being processed simultaneously.
-1	"ARRFULL"	IFREE-NXTENT is less than 6, so that there is insufficient space for entry of the 6 table items in array IGRPH. IFREE: pointer to free block for table entries. NXTENT: pointer to free block for linked list entries. <u>Probable cause:</u> System fault. Reduce the number of subprograms being processed simultaneously to prevent IGRPH overflow

CHR

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1) contains a character C	The character C is in error, being neither a blank, letter, nor digit. <u>Probable cause:</u> An illegal FORTRAN character has been encountered.

CMDOWN

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"MISSING BLOCK"	There should be an entry for the common block in IGRPH for this routine, yet it is missing. <u>Probable cause:</u> System fault.
-2	"UNKNOWN AVAIL"	The adjusted availability classification of a common block in the calling routine being processed cannot be determined on the basis of the combined classifications in the calling and called routines; one of these classifications must be in error. <u>Probable cause:</u> System fault.

COMTCH

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=IPR(1) (2)= (2)	There is no entry for the COMMON block in IGRPH. <u>Probable cause:</u> System fault.
-2	ERRPKT(1)=IPR(1) (2)= (2)	The block is always available in the called subprogram and either sometimes or never available in the calling. <u>Probable cause:</u> System fault. [IPR[1]] is the name of an external reference in the subprogram being

COMTCH

Continued

NO.MESSAGEMEANING

processed] IPR[2] is the name of a
COMMON block in IPR[1] currently being
processed]

CONTOK

NO.MESSAGEMEANING

-1	"BAD TYPE"	The type code (ITYPE) from the array TKSTK is outside the permissible range. Specifically, either: ITYPE<JLBLE or ITYPE>JENDS. <u>Probable cause:</u> System fault.
-2	"NO END"	INDEX should point to the next type code in TKSTK. It is found that INDEX equals or exceeds the length (LAST) of TKSTK. <u>Probable cause:</u> System fault.
-3	"BAD IN"	The starting point (ISTART) of the string input is found to exceed what should be the end (IEND), indicating an error in TKSTK. <u>Probable cause:</u> System fault.
-4	"BAD IN"	The size (IFNIN) of the array INSTK is less than the index (IEND) of the last character of the string input. <u>Probable cause:</u> System fault.
-5	"BAD IN"	The end (IFNIN) of the array INSTK has been reached (on an apparent ASSIGN statement) without finding a marker "T" to signify the end (IEND) of the input string. <u>Probable causes:</u> System fault, or syntax error in ASSIGN TO statement.

DATYP

F.

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"KEY ERROR"	The statement being processed is not a type declaration statement as expected; either IVAL<KIN or IVAL>KLG. [IVAL contains the type code of the statement.]

Probable cause: System fault.

ENPATH

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=INDEX	There is no entry for this variable in any list of parameters or common variables in the subprogramwide table.

EQCHN

F.

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"EQUIV ERR"	The index (ICNX) for table ICHTB exceeds its maximum allowable value (IQMAX). The system places an upper limit of 100 on the number of variables which can be made (pairwise) equivalent. <u>Probable cause:</u> System fault.
-2	"EQUIV ERR"	<u>Probable cause:</u> The same variable or array has been incorrectly equivalenced twice in an illegal manner.

EQUIV

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"IQMAX"	The number of entries (IQLN) in IQLST exceeds the maximum number allowed (IQMAX). <u>Probable cause:</u> System fault. The system places an upper limit of 100 on the number of variables which can be made (pairwise) equivalent.

EXPRSN

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=IBEG (2)=IEND	Source string indices are incorrect: IBEG>IEND <u>Probable cause:</u> System fault.
-2	ERRPKT(1)=NEXTPT (2)=LENSTT	The next transition is in error; NEXTPT no longer points to the transition table. (NEXTPT must be less than the length of the table LENSTT). <u>Probable cause:</u> Syntax error in an expression.
-3	ERRPKT(1)=NXTST (2)=KRSTA	The next state is in error; NXTST must be one of 1, 2, 3, 4, 5, but has been found to be 0. <u>Probable cause:</u> Syntax error in an expression. One possibility is the illegal juxtaposition of 2 operators.

FCALL

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"STACKVALUEERROR"	A parameter classification of JPREX or JASFA was expected and is not found. <u>Probable cause:</u> System fault.
-2	"TOKEN ERROR"	The token (ITOK) is in error, being none of the expected code number identifiers. <u>Probable cause:</u> System fault.
-3	"UNDERFLOW"	An underflow condition exists, IPT (the index to the array IPAR) is less than 0, or IP (the pointer to the array ISTK) is less than 0. <u>Probable causes:</u> System fault, or incorrect parentheses count in source code: in subprogram or arithmetic statement function call.

FLTCHR

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1),..., ERRPKT(I2-I1+1) contains the segment V(I1),..., V(I2)	The input segment V(I1),...,V(I2) does not contain a valid representation of a real number. <u>Probable cause:</u> Syntax error in Fortran representation of a real number.
-1	ERRPKT(1)=I1 ERRPKT(2)=I2 ERRPKT(3)=V(I1)	The parameters I1, I2 of the input string are in error, since I2>I1. <u>Probable cause:</u> System error.

GRORDR

F.

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"BADNAME"	The array IGRPH, which contains calling-tree information, does not contain the name of the caller "N" of the subprogram being processed, yet IGRPH indicates that the subprogram is called by "N" <u>Probable cause:</u> System fault.

INPVAR

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"UNDERFLOW"	An underflow condition exists. The index (IP) into array ISTK is less than 0. <u>Probable cause:</u> System fault.
-2	"OVERFLOW"	An overflow condition exists. The index (IP) into array ISTK exceeds its maximum allowable value (LIM). This value is supplied in a data statement. <u>Probable cause:</u> Systems fault.
-3	"UNDEFINED INPUT"	The I/O classification for the input variable being examined fits no legal category. <u>Probable cause:</u> Systems fault.

INTCHR

-1	ERRPKT(1),..., ERRPKT(I2-I1+1) contains the segment V(I1),..., V(I2)	The input segment V(I1),...,V(I2) does not contain a valid representation of an integer. <u>Probable causes:</u> Syntax error in the representation of an integer, or system fault.
-1	ERRPKT(1)=I1 ERRPKT(2)=I2 ERRPKT(3)=V(I1)	The parameters I1, I2 of the input string are in error, since I2>I1. <u>Probable cause:</u> System error.

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"PASTEND"	An overflow condition exists; an index (INNOD) into array IBBLT exceeds the maximum allowable value (IBBLTL). <u>Probable cause:</u> System fault. <u>Possible user correction:</u> reduce the number of basic blocks or the number of local or global variables.
-2	"PASTEND"	An overflow condition exists; an index (ISTP) into array ISTOP exceeds the maximum allowable value (ISTPL). <u>Probable cause:</u> System fault. <u>Possible user correction:</u> reduce the number of basic blocks or the number of local or global variables.
-3	"PASTEND"	An overflow condition exists; an index (NUM) into array IBIND* equals or exceeds the value IBNDL. <u>Probable cause:</u> System fault.
-4	"UNCLASS"	The variable's I/O status is unclassified. <u>Probable cause:</u> System fault.
-5	"PASTEND"	An overflow condition exists; an index (IDTEND) into array IBBLT exceeds the allowable value (IBBLTL). <u>Probable cause:</u> System fault.
-6	"PASTEND"	More than IELMT(=100) variables are equivalenced in a subprogram, exceeding the system limit. <u>Probable cause:</u> system fault.
-7	"PASTEND"	More than INIDL(=200) global or local variable appear in a subprogram, exceeding the system limit.

IOSTAT

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"CODE ERR"	The I/O class of a parameter in a previously processed subprogram is incorrect. <u>Probable cause:</u> System fault.

IOVUND

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=IEXITB (2)=NUM (3)=LHEAD (4)=KCOUNT	Block IEXITB does not have a linked list of entry blocks. It should have a linked list with at least NUM as an entry block. <u>Probable cause:</u> System fault.
-2	ERRPKT(1)=IEXITB (2)=NUM (3)=LHEAD (4)=KCOUNT (5)=LLNUM	Block IEXITB does not have NUM in its entry block list, although IEXITB is listed as an exit block for NUM. <u>Probable cause:</u> System fault.
-3	ERRPKT(1)=KCOUNT	KCOUNT exceeds KSIZE(=50) in value; 50 is the system limit on the possible number of undefined output variables.

ISTYPE

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=ICHAR (2)=IPONT	A non-ANSI character (ICHAR) is found, where ICHAR=INSTK(IPONT). <u>Probable cause:</u> Syntax error. Check source for the use of an illegal character.

LABBLK

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=K (2)=LHEAD (3)=LSAVE	The linked list of internal references to an external label is in error. Either: (1) the linked list was constructed improperly so that the list never points back to its original header node, or, (2) there are more than INLABX elements in the linked list. LSAVE: ptr to header for the linked list of internal ref's to an external label. LHEAD: ptr to current node in the linked list. K: block number of external label whose references are being sought. <u>Probable cause:</u> System fault.
-2	ERRPKT(1)=NUMDO ERRPKT(2)=KDO	NUMDO exceeds KDO(=20). The system places an upper limit of 20 on the number of DO statements referring to the same DO-loop closure statement. <u>Probable cause:</u> System fault. <u>Possible user correction:</u> Alter DO statements to correct the above situation.
-3	ERRPKT(1)=JT ERRPKT(2)=INLABX	JT exceeds INLABX(=50). The system places an upper limit of 50 on the number of statements which refer to the same external label. <u>Probable cause:</u> System fault. <u>Possible user correction:</u> Alter statement referencing to correct the above situation.

LASTNB

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=IBEG (2)=IEND (3)=IPONT	The pointer (IPONT) is out of range. Must have: IBEG less than or equal to IPONT less than or equal to IEND. <u>Probable cause:</u> System fault.

LEXST

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
IERR = +1	ERRPKT(1)=IERR (2)=J (3)=0 (4)=LLAST (5)=JTYPE (6)=JCODE	The beginning of the subprogram or block data was reached, and no executable statement was found. JCODE=1: subprogram search 2: block data search. <u>Probable cause:</u> System fault.
+2	ERRPKT(1)=IERR (2)=J (3)=JBLOCK (4)=LLAST (5)=JTYPE (6)=JCODE	The beginning of the block was reached and no executable statement was found. <u>Probable cause:</u> System fault.
+3	ERRPKT(1)=IERR (2)=J (3)=0 (4)=LLAST (5)=JTYPE (6)=JCODE	The value of JCODE was neither 1 nor 2 as required. <u>Probable cause:</u> System fault.

LUNTIL

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	[none]	No one of the specified characters being sought was found. <u>Probable cause:</u> Syntax error. Check source code for missing parentheses in an IF statement or for a missing character [0] in an ASSIGN TO statement.

MKLBTB

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=TKSTK(1) (2)= (2) (2*LCLEN+6)	There is a labelling error in an ASSIGN, GOTO, or DO statement. <u>Probable cause:</u> Syntax error. (2*LCLEN) +6

NAMCHR

-1	ERRPKT(1),..., ERRPKT(I2-I1+1) contains the segment V(I1),..., V(I2)	The input segment V(I1),...,V(I2) does not contain a valid representation of a FORTRAN variable name. <u>Probable causes:</u> Syntax error in FORT. representation of a variable name, or system fault.
-1	ERRPKT(1)=I1 ERRPKT(2)=I2 ERRPKT(3)=V(I1)	The parameters I1, I2 of the input string are in error, since I2>I1. <u>Probable cause:</u> System error.

NEXTNB

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=IBEG (2)=IEND (3)=IPONT	The pointer (IPONT) is out of range. Must have: IBEG-1 less than or equal to IPONT less than IEND. <u>Probable cause:</u> System fault.

OUTVAR

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"UNDERFLOW"	The index (IP) into array ISTK is less than 0. <u>Probable cause:</u> System fault.
-2	"OVERFLOW"	The index (IP) into array ISTK exceeds its maximum allowable value (LIM). This value is supplied in a DATA statement. <u>Probable cause:</u> System fault.
-3	"OVERFLOW"	The index (IST) into the array ISTOP, the list of stop nodes, exceeds its maximum allowable value (ISTL). <u>Probable cause:</u> System fault.
-4	"UNDEF"	The I/O classification for the output variable being examined fits no legal category. <u>Probable cause:</u> System fault.

PHASE1

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	[none]	<p><u>Probable cause:</u> Syntax error. Check for the following:</p> <p>(a) a missing (non-blank) character</p> <p>(b) an error in an ASSIGN statement; "0" not found.</p> <p>(c) an error in a DO statement: required variable not found.</p>
-2	IERR(1) contains (illegal) value of ITYPE	<p>An illegal statement type has been encountered.</p> <p><u>Probable cause:</u> Syntax error. Check for a non-ANSI FORTRAN statement.</p>

PRACI

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT contains a list of parameters	<p>The number of parameters in an arithmetic statement function definition exceeds the system limit IDLMT (=20).</p> <p><u>Probable cause:</u> System fault.</p>

PRODM

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"NOTDM"	A dimension statement was expected and not found (ITOK<0).
-2	"NOLFTPAR"	<p><u>Probable causes:</u> Either a system fault, or a syntax error in a dimension statement. Check for a missing left parenthesis.</p>

PUSH

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT=ISTK	<p>An overflow condition exists: IP>LSTK where LSTK=100 is the max value of the 1st index in the array ISTK.</p> <p><u>Probable cause:</u> System fault</p> <p>ISTK, created by FCALL, contains the type of call being analyzed in order to handle nested calls.</p>

PUTTOK

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=IALPHA (2)=IBETA (3)=IFNTK (4)=LNTOK	<p>An overflow condition exists. The index (IFNTK) to the token stack (TOKST) equals or exceeds the max allowable dimension, LNTOK minus 1.</p> <p><u>Probable cause:</u> System fault.</p>

RDWR

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	IM.GT.20	<p>An overflow condition for array IND exists; IND is dimensioned IND(20,2), and the 1st subscript, referred to as IM, exceeds 20.</p> <p><u>Probable cause:</u> System fault.</p> <p><u>Possible correction:</u> Change source breaking up READ or WRITE statement into several statements.</p>

RDWR

Continued

-3 "NOT IDENT"

The system allows a maximum of 50.

The variable being classified is neither a scalar nor an array; it is not a proper identifier.

Probable cause: Error in source - something other than a scalar or array appears in a READ or WRITE statement.

SAVEIO

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-2	"VARDIM"	A COMMON array is found to be variably dimensioned; IVAR=1 is returned by the call to BSUBTB. <u>Probable cause:</u> Use of non-ANSI FORTRAN.

SBMTCH

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"OVERFLOW"	The index (IND) into array IGRPH exceeds the max. allowable value (NXTENT). <u>Probable cause:</u> System fault. There is a missing entry in IGRPH.
-2	"OVERFLOW"	The index (II) into array ICOM exceeds the max. allowable value, 60. <u>Probable cause:</u> System fault. The number of COMMON blocks sometimes or always available to a subprogram cannot exceed 30.

SKIP

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	[none]	The number of non-blank characters designated could not be found. <u>Probable cause:</u> Syntax error. Check for a non-ANSI FORTRAN statement.

STEPAR

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"PAR ERR"	(KIND<IHND) An argument in a call is not of allowable type. <u>Probable causes:</u> Either a syntax error or a system fault.

STRDM

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	"BADTOK"	The value (ITOK) of the token is at variance with the expectations for a dimension statement, since ITOK indicates neither: a comma, right parenthesis, left parenthesis, variable name, nor integer. <u>Probable cause:</u> Check for a syntax error in a dimensioned variable appearing in a declaration statement.

STRDM

Continued

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-2	"NORTPAR"	<u>Probable cause:</u> Syntax error - the right parenthesis is missing for a dimensioned variable.

TRANS

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
-1	ERRPKT(1)=ICODE (2)=KA (3)=KB	The transfer from block KA to KB is not of a recognizable type, i.e., ICODE<1 or ICODE>8 where ICODE indicates the transfer type. <u>Probable cause:</u> System fault.
-2	ERRPKT(1)=ICODE (2)=KA (3)=KB	It is found that LOCA = 0, yet LOCA should be a pointer to the node when KB is an exit block for block KA. <u>Probable cause:</u> System fault.
-3	ERRPKT(1)=ICODE (2)=KA (3)=KB	It is found that LOCB = 0, yet LOCB should be a pointer to the node where KA is an entry block to block KB. <u>Probable cause:</u> System fault.
-4	ERRPKT(1)=ICODE (2)=KA (3)=JTYPE	The type JTYPE of transfer from block KA to block KB is not as expected. <u>Probable cause:</u> System fault.

VARPR

<u>NO.</u>	<u>MESSAGE</u>	<u>MEANING</u>
+1	ERRPKT(1)=INPVAR (2)= J	The input variable (INPVAR) is in a basic block output list with I/O type undefined. Thus the variable has no value assigned to it for use as input. J = line number of the statement containing the problem. <u>Probable cause:</u> Syntax error. Check for illegal use of a variable in line J of code.