LABEL 000000000PRNT 00177279?USER=SPO ; EXECUTE COPY /SPO COPY /SPO

FILE: SYSTEM/ BULLTIN THURS 10/06/77 11:03AM

A-A-A

ECU COMPUTER CENTER

B-5500 SYSTEMS BULLETIN INDEX

OCTOBER 1, 1976

BULLETIN NUMBER	SUBJECT	NUMBER OF REVISIONS
01	MONITOR STATEMENT IN COBOL68	1
02	USING/GIVING SORTS IN COBOL68	
03	CORRESPONDING OPTIONS IN COBOL68	1
04	TABLE SIZE LIMITS IN COBOL68	1
05	FORTRAN OBJECT PROGRAM TERMINATION MESSAGES	
06	READ ERROR BRANCH IN FORTRAN	
07	ZERO SUPPRESSION IN COBOL68	1
08	OPTIMUM USE OF CONSTANTS IN COBOL68	
09	THE "IF NUMERIC" TEST IN COBOL68	
10	SORT PARAMETER DEFAULT VALUES IN COBOL68	
11	PARITY CHECKING IN COBOL68	1
12	EBCDIC OPTION FOR BASIC COMPILER	
13	RELEASE 16.0 IMPLEMENTATION - COMPILERS	
14	RELEASE 16.0 IMPLEMENTATION - FORTRAN	
15	MAXIMUM NUMBER OF ELEMENTS IN FORTRAN DATA STATEMENT	
16	RELEASE 16.0 IMPLEMENTATION - MCP AND INTRINSICS	
17	COMPUTATIONAL ITEMS IN COBOL68	2
18	CHANGES AND NEW FEATURES IN 16.0 - BATCH MCP	
19	CHANGES AND NEW FEATURES IN 16.0 - TIME-SHARING MCP	
20	CHANGES AND NEW FEATURES IN 16.0 - XALGOL	
21	CHANGES AND NEW FEATURES IN 16.0 - BASIC	
22	CHANGES AND NEW FEATURES IN 16.0 - COBOL68	
23	CHANGES AND NEW FEATURES IN 16.0 - FORTRAN	
24	DOCUMENTATION FOR NEW AND IMPROVED FORTRAN FEATURES	
25	SUBSCRIPTING IN COBOL68	1
26	GUIDELINES FOR TAPE AND DISK FILE ATTRIBUTES	
27	NOTES ON FILE EQUATION	
28	SEGMENTATION IN FORTRAN	
29	COPY VERB IN COBOL68	
30	COBOL68 SORT PROBLEM	1
31	COBOL68 RESERVED WORDS	
32	XALGOL WRITE STATEMENT	

EAST CAROLINA UNIVERSITY
COMPUTING CENTER

MEMORANDUM

TO: DISTRIBUTION

FROM: CARL B. FRIEDLANDER
SYSTEMS PROGRAMMER

DATE: APRIL 1, 1976

SUBJECT: REVISION OF THE B5500 SYSTEMS BULLETINS

THE USER SERVICES GROUP OF THE COMPUTING CENTER, OF WHICH I AM A MEMBER, HAS BEEN STUDYING THE CONCEPT OF THE SYSTEMS BULLETIN FOR THE PAST TWO WEEKS. WE HAVE CONCLUDED THAT THE SYSTEMS BULLETIN IS A SOUSD AND FEASIBLE MEANS OF COMMUNICATION BETWEEN THE SYSTEMS PROGRAMMER AND THE USERS. WE STUDIED THE FORMAT OF THE DOCUMENT AND FOUND IT TO BE GENERALLY SATISFACTORY. WE DID FEEL, HOWEVER, THAT SOME WORK WAS IN ORDER CONCERNING UPDATE AND ORGANIZATION.

AS A RESULT OF OUR STUDY, SOME CHANGES WERE IMPLEMENTED IN THE AREA OF OVERALL ORGANIZATION. EACH SYSTEM BULLETIN THAT HAD BEEN ISSUED (THERE WERE 18) WAS REVIEWED. EVERY ONE THAT DEALT WITH MULTIPLE SUBJECTS WAS DIVIDED INTO SEPARATE BULLETINS FOR EACH SUBJECT. THIS DIVISION NECESSITATED THE RE-NUMBERING OF ALL OF THE BULLETINS IN ORDER TO MAINTAIN A CHRONOLOGICAL ORDER. FURTHERMORE, BULLETINS WHICH PROVIDE NOTIFICATION OF SOME MALFUNCTION IN SOFTWARE OR NOTIFICATION OF A CORRECTION TO A MALFUNCTION ARE NOW DESIGNATED AS SUCH IN THE SUBJECT LINE OF THE BULLETIN. UPDATE OR CHANGES TO A BULLETIN WILL NO LONGER BE ISSUED AS A SEPARATELY NUMBERED BULLETIN BUT INSTEAD WILL BE RELEASED WITH THE ORIGINAL BULLETIN NUMBER AND A REVISION LEVEL. THIS WILL FACILIATE ATTACHMENT OF ALL UPDATES TO THE BULLETIN TO WHICH THEY ARE APPLICABLE.

PLANS FOR A NEW DISTRIBUTION POLICY ARE BEING FORMULATED; MEANWHILE, NEW BULLETINS WILL BE POSTED ON THE BULLETIN BOARD IN AUSTIN-108. USER SERVICES WILL MAINTAIN, IN THE DOCUMENTATION LIBRARY, A BOOK CONTAINING TWO COMPLETE SETS OF SYSTEMS BULLETINS; ONE SET WILL BE MAINTAINED IN NUMERICAL ORDER BY BULLETIN NUMBER AND THE OTHER WILL BE MAINTAINED BY SUBJECT AREA. AS HAS BEEN DONE IN THE PAST, NEW BULLETINS ISSUED DURING A MONTH WILL BE SUMMARIZED IN THE NEXT MONTH"S NEWSLETTER.

A NEW AND COMPLETE SET OF THE FIRST 25 SYSTEMS BULLETINS IS ENCLOSED WITH THIS MEMORANDUM. YOUR ENTIRE SET OF OLD BULLETINS SHOULD BE DESTROYED AND REPLACED WITH THIS NEW SET. A FEW OF THE ORIGINAL

A-A-A

BULLETINS ARE BEING RE-WRITTEN AND WILL BE RELEASED IN A FEW DAYS.

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 1, REVISION 1

02 APRIL 76

SUBJECT: REVISION NOTICE - MONITOR STATEMENT IN COBOL68
-------- -------- ------ - ------- --------- -- -------

THE KNOWN PROBLEMS WITH THE MONITOR STATEMENT HAVE BEEN CORRECTED. THE CORRECTION IS EFFECTIVE IN COBOL68 LEVEL 16.0.13 AND ALL SUBSEQUENT LEVELS.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*  THIS SHOULD BE A PERMANENT ATTACHMENT TO THE ORIGINAL BULLETIN    *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 1

9 APRIL 75

SUBJECT: ERROR NOTIFICATION - MONITOR STATEMENT IN COBOL68
-------- ----- ------------ - ------- --------- -- -------

AVOID THE USE OF THE MONITOR STATEMENT IN COBOL68 UNTIL FURTHER NOTICE. UNDER CERTAIN CONDITIONS THE MONITOR STATEMENT ALTERS THE VALUE OF DATA-NAMES BEING MONITORED.

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 2

15 APRIL 75

SUBJECT: ERROR CORRECTION - USING/GIVING SORT IN COBOL68
-------- ----- ---------- - ------------ ---- -- -------

A PATCH HAS BEEN INSTALLED IN BOTH THE BATCH AND TSS I-TRINSICS TO ALLOW A COBOL68 USING/GIVING SORT TO FUNCTION PROPERLY WHEN BOTH THE USING AND GIVING CLAUSES REFFRENCE THE SAME FILE (SAMF FD NAME). THIS BECAME EFFECTIVE AT 9 A.M. ON MONDAY, APRIL 14, 1975. PRIOR TO THIS PATCH A USING/GIVING SORT REFERENCING THE SAME FILE FOR INPUT AND OUTPUT WOULD BOMB OUT WITH AN -INVALID ADDRESS OR -STACK OVERFLOW CONDITION.

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 3, REVISION 1

11 SEPTEMBER 75

SUBJECT: REVISION NOTICE - CORRESPONDING OPTION IN COBOL68
-------- -------- ------ - ------------- ------ -- -------

THE KNOWN PROBLEMS WITH THE USE OF "CORRESPONDING" IN CONJUNCTION WITH SUBSCRIPTED DATA ELEMENTS HAVE BEEN CORRECTED. THE CORRECTION IS EFFECTIVE IN THE COBOL68 COMPILER LEVEL 16.0.02 AND ALL SUBSEQUENT LEVELS.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*  THIS SHOULD BE A PERMANENT ATTACHMENT TO THE ORIGINAL BULLETIN   *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 3

23 APRIL 75

SUBJECT: ERROR NOTIFICATION - CORRESPONDING OPTIONS IN COBOL68

AVOID THE USE OF MOVE CORRESPONDING, ADD CORR5SPONDING, AND SUBTRACT CORRESPONDING WHEN EITHER THE SENDING OR RECEIVING FIELDS ARE SUBSCRIPTED. THE SUBSCRIPTS ARE NOT HANDLED CORRECTLY AND INCORRECT RESULT2 WILL OCCUR.

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 4, REVISION 1

05 APRIL 76

SUBJECT: REVISION NOTICE - TABLE SIZE LIMITS IN COBOL68

THE PROBLEM WHEREBY A SYNTAX ERROR WAS NOT PRODUCED IF AN ARRAY EXCEEDED THE LIMIT HAS BEEN CORRECTED. THE CORRECTION IS EFFECTIVE IN COBOL68 LEVEL 16.0.04 AND SUBSEQUENT LEVELS.

```
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *  THIS SHOULD BE A PERMANENT ATTACHMENT TO THE ORIGINAL BULLETIN   *
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 4

23 APRIL 75

SUBJECT: ERROR NOTIFICATION - TABLE SIZE LIMITS IN COBOL68

IN COBOL68 THE MAXIMUM SIZE OF ANY TABLE IS 8184 CHARACTERS. IN SOME CASES TABLES DECLARED IN EXCESS OF THIS LIMIT WILL NOT BE FLAGGED AS BEING IN ERROR UNTIL ALL OTHER ERRORS ARE CORRECTED.

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 5

28 APRIL 75

SUBJECT: FORTRAN OBJECT PROGRAM TERMINATION MESSAGES

THE RECORD NUMBER WHICH IS OUTPUT IN THE FORTRAN OBJECT PROGRAM TERMINAL MESSAGE SUCH AS

```
--TYPE ERR ON FILE . . .
--NAMELIST ERROR ON FILE . . .
```

AND OTHER SIMILAR MESSAGES REFERS TO THE PHYSICAL RECORD NUMBER (BLOCK NUMBER). IF THE FILE IS UNBLOCKED, THE PHYSICAL RECORD NUMBER WILL BE THE SAME AS THE LOGICAL RECORD NUMBER; HOWEVER, IF THE FILE IS BLOCKED AND THE ERROR IS ON RECORD #N, LOOK IN THE NTH BLOCK TO FIND THE ERROR

RATHER THAN IN THE NTH RECORD.
A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 6

20 MAY 75

SUBJECT: READ ERROR BRANCH IN FORTRAN

ALTHOUGH THE "ERR=" OPTION OF A FORTRAN READ STATEMENT IS PRIMARILY DESIGNED TO ALLOW PROGRAMMATIC HANDLING OF PARITY ERRORS, IT MAY BE USED TO PROGRAMMATICALLY PROCESS TYPE ERRORS ON INPUT. ANY TYPE ERROR OR PARITY ERROR WILL CAUSE A BRANCH TO BE TAKEN TO THE LABEL INDICATED BY THE "ERR=" SPECIFICATION. THE RECORD MAY THEN BE PROCESSED IN ANY MANNER DESIRED, INCLUDING RE-READING THE RECORD WITH ANOTHER FORMAT. NOTE, HOWEVER, THAT SOME SORT OF A READ SOATEMENT MUST BE EXECUTED IF ONLY A DUMMY READ TO CLEAR THE BUFFER. IF NO READ IS EXECUTED, THE SYSTEM WILL INFINITELY READ THF OFFENDING CARD.
A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 7, REVISION 1

05 APRIL 76

SUBJECT: REVISION NOTICE - ZERO SUPPRESSION IN COBOL68

THE KNOWN PROBLEMS WITH ZERO SUPPRESSION HAVE BEEN CORRECTED. THE CORRECTION IS EFFECTIVE IN COBOL68 LEVEL 16.0.02 AND ALL SUBSEQUENT LEVELS.

* *

* THIS SHOULD BE A PERMANENT ATTACHMENT TO THE ORIGINAL BULLETIN *

* *

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 7

20 MAY 75

SUBJECT: ERROR NOTIFICATION - ZERO SUPPRESSION IN COBOL68

MOVING A DATA-NAME WITH A VALUE OF ZERO TO A DATA-NAME WITH TOTAL ZERO SUPPRESSION EDITING (I.E., PICTURE OF ALL Z"S OR PICTURE CONTAINING BLANK WHEN ZERO CLAUSE) MAY CAUSE A PROGRAM OR SYSTEM HANG. AVOID THE USE OF TOTAL ZERO SUPPRESSION WHENEVER POSSIBLE UNTIL FURTHER NOTICE.

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 8

18 JUNE 75

SUBJECT: OPTIMUM USE OF CONSTANTS IN COBOL68

CONSTANTS IN THE RANGE -1023 ≤ CONSTANT ≤ 1023 SHOULD BE USED IN THE FORM OF NUMERIC LITERALS FOR OPTIMUM RESULTS IN COBOL68; HOWEVER, CONSTANTS OUTSIDE THIS RANGE SHOULD BE GIVEN A LEVEL 77 COMP-1 DATANAME FOR OPTIMUM RESULTS.

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 9

7 JULY 75

SUBJECT: THE "IF NUMERIC" TEST IN COBOL68

NOTE THAT THE NUMERIC TEST ON AN ELEMENT DECLARED WITH X"S AS ITS PICTURE WILL NOT ALLOW A SIGNED NUMBER. THE RESULT WILL BE NUMERIC ONLY IF EACH DIGIT OF THE ELEMENT CONTAINS A NUMBER. IF A SIGNED NUMBER IS TO BE ALLOWED, THEN THE X-FIELD MUST BE REDEFINED AS A 9-FIELD WITH A SIGN INDICATOR. THE NUMERIC TEST SHOULD THEN BE DONE ON THE REDEFINED AREA. DO NOT MOVE THE X-FIELD TO A 9-FIELD TO ACCOMPLISH THIS AS THIS WILL CAUSE A STRIPPING OF THE ZONE BITS AND THE DATA WILL ALWAYS BE NUMERIC.

EXAMPLES:

VALUE	PICTURE	RESULT
11J (-111)	XXX	NOT NUMERIC
11J	S999 (REDEFINES)	NUMERIC
1J1	XXX	NOT NUMERIC
1J1	S999 (REDEFINES)	NOT NUMERIC
K22 (-222)	XXX	NOT NUMERIC
K22	J999 (REDEFINES)	NUMERIC
111	XXX	NUMERIC
111	J999 (REDEFINES)	NUMERIC
111	S999 (REDEFINES)	NUMERIC

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 10

7 JULY 75

SUBJECT: SORT PARAMETER DEFAULT VALUES IN COBOL68

1. DUE TO AN OVERSIGHT, THE DEFAULT OPTIONS FOR MEMORY SIZE AND DISK SIZE (BOTH PERTAINING TO SORT) HAVE BEEN INCORRECT. THESE ARE NOW CORRECT AND AS FOLLOWS:

   MEMORY SIZE = 6000 WORDS
   DISK SIZE = .25 MODS.

   THE DEFAULT OPTIONS MAY BE OVERRIDDEN BY INCLUSION OF THE PROPER PHRASES IN THE OBJECT-COMPUTER CLAUSE.

2. SORTS SHOULD NOT BE DIRECTED TO USE DISK AND TAPES WHEN ONE DOES NOT INTEND TO USE TAPES. THE DISK-ONLY SORT SHOULD BE USED UNLESS LARGE VOLUMES OF DATA ARE TO BE SORTED. ASSIGNING SMALL SORTS TO A TAP& AND DISK SORT SOMETIMES CAUSES PROBLEMS BY SWITCHING TO TAPE WHEN IT IS NOT NECESSARY.

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 11, REVISION 1

06 APRIL 76

SUBJECT: REVISION NOTICE - PARITY CHECKING IN COBOL68

THE COBOL68 PROCEDURE "PARITY/CHECK" WILL NOT BE AVAILABLE AFTER 9:00 AM ON THURSDAY, APRIL 8, 1976. PARITY CHECKING IS NOW HANDLED AUTOMATICALLY BY THE MCP. PROGRAMMATIC PARITY HANDLING IS NECESSARY ONLY IF THE PROGRAM IS TO PROCESS THE BAD DATA.

```
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *  THIS SHOULD BE A PERMANENT ATTACHMENT TO THE ORIGINAL BULLETIN   *
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 11

11 JULY 75

SUBJECT: PARITY CHECKING IN COBOL68

A STANDARD PROCEDURE FOR PARITY CHECKING IN COBOL68 IS NOW AVAILABLE. THE PROCEDURE, WHICH REMAINS ON DISK AS A PERMANENT FILE, MAY BE INCLUDED IN A COBOL68 PROGRAM BY INCLUDING THE FOLLOWING CARD AS THE FIRST STATEMENT OF THE PROCEDURE DIVISION.

```
COPY "PARITY/CHECK" REPLACING BOMB BY FILE-NAME.
```

WHERE FILE-NAME IS THE NAME OF ANY FILE (OTHER THAN A PRINT FILE) IN THE PROGRAM WHICH REMAINS OPEN DURING THE P9OGRAM.

THE PARITY CHECK ROUTINE WILL TYPE AN APPROPRIATE MESSAGE TO THE OPERATOR AND THEN DO AN OPEN ON THE SPECIFIED FILE. SINCE THIS FILE IS ALREADY OPEN, THE PROGRAM WILL BE DS-ED WITH AN IO ERROR.

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 12

31 JULY 75

SUBJECT: EBCDIC OPTION FOR BASIC COMPILER

THE BASIC COMPILER NOW HAS AN OPTION SIMILAR TO THE H6L OPTION IN FORTRAN FOR PROVIDING EBCDIC TO BCL CONVERSION. THIS OPTION WILL PROVIDE FOR PROPER CHARACTER TRANSLATION OF BASIC PROGRAMS WHICH ARE PUNCHED ON IBM-029 KEYPUNCHES. THE EBCDIC OPTION DEFAULTS TO RESET BUT MAY BE SET AND RESET IN THE STANDARD MANNER USED FOR OTHER $-CARD OPTIONS.

EXAMPLES:

```
$ CARD LIST EBCDIC

$ SET EBCDIC
```

EBCDIC CHARACTER SET FOR BASIC ON THE B5500

A TO Z	.	(	@
0 TO 9	,	)	&
BLANK	"	<	$
+	;	>	=
-	*	#	%
/			

WARNING * * * REFER TO APPENDIX B OF THE BASIC MANUAL WHEN USING BASIC THROUGH THE CARD READER.

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 13

26 JULY 75

SUBJECT: RELEASE 16.0 IMPLEMENTATION - COMPILERS

IMPLEMENTATION OF SOFTWARE RELEASE MARK 16.0 HAS BEGUN ON SCHEDULE. THE RELEASE IS BEING PREPARED AND BROUGHT ONLINE IN PHASES AS HAS BEEN THE CUSTOM IN THE PAST.

THE NEW RELEASE DOES CONTAINS SEVERAL NEW FEATURES FOR BOTH THE MCP AND COMPILERS. SOME ADDITIONAL LOCAL ENHANCEMENTS ARE BEING INSTALLED CONCURRENTLY, ALSO. ALTHOUGH EVERY EFFORT IS BEING MADE TO KEEP PROBLEMS TO A MINIMUM, IT IS POSSIBLE THAT A FEW MINOR PROBLEMS MAY BE EXPERIENCED. USERS ARE ENCOURAGED TO BE ON THE LOOKOUT FOR ANY PROBLEMS A5D TO AID US BY REPORTING ANY PROBLEMS EXPERIENCED.

TO AVOID CONFUSION, THE NEW IMPLEMENTATIONS WILL NOT BE ANNOUNCED AT THIS TIME. AS SOON AS THE INITIAL IMPLEMENTATION OF THE ENTIRE RELEASE IS COMPLETE, ALL CHANGES, IMPLEMENTATIONS AND ENHANCEMENTS WILL BE RELEASED IN DETAIL.

THE FOLLOWING RELEASE 16 SOFTWARE WAS BROUGHT ONLINE AT 9:00 AM ON 8/25/75: ALGOL, XALGOL, BASIC, COBOL, COBO368, CANDE.

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 14

28 AUGUST 75

SUBJECT: RELEASE 16.0 IMPLEMENTATION - FORTRAN

THE RELEASE 16 FORTRAN COMPILER WAS BROUGHT ONLINE AT 9:00 AM ON 8/28/75. THERE ARE NO OFFICIAL ENHANCEMENTS TO FORTRAN; HOWEVER, THERE ARE SEVERAL LOCAL ENHANCEMENTS. DETAILS WILL BE RELEASED AT A LATER DATE.

USERS ARE ENCOURAGED TO BE ON THE LOOKOUT FOR ANY PROBLEMS AND TO AID US BY REPORTING ANY PROBLEMS EXPERIENCED.

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 15

8 SEPTEMBER 75

SUBJECT: MAXIMUM NUMBER OF ELEMENTS IN FORTRAN DATA STATEMENT

THE MAXIMUM NUMBER OF DATA-ITEMS THAT MAY BE INCLUDED IN A SINGLE DATA STATEMENT IN FORTRAN IS 128, WHERE A DATA-ITEM IS OF THE FORM R* CONSTANT. THE COMPILER IS SMART ENOUGH, HOWEVER, TO SQUASH LISTS OF ADJACENT CONSTANTS INTO THAT FORM WHERE POSSIBLE.

```
    EXAMPLE:
                 DATA I/1,2,3,4,5,5,5,5,5,5,6,7,7/
    WHERE I IS AN ARRAY IS CONSIDERED TO BE
                 DATA I/1,2,3,4,6*5,6,2*7/
    AND THUS CONTAINS 7 DATA-ITEMS RATHER THAN 13
```

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 16

07 OCTOBER 75

SUBJECT: RELEASE 16.0 IMPLEMENTATION - MCP AND INTRINSICS

THE FINAL PHASE OF RELEASE 16.0 INSTALLATION WILL BEGIN AT 9:00 AM ON MONDAY, OCTOBER 13, 1975. AT THIS TIME, THE MCP (BOTH BATCH AND TIME-SHARING) AND INTRINSICS WILL BE BROUGHT ONLINE FOR FINAL TESTIN7. BARRING ANY COMPLICATIONS, THIS RELEASE WILL REMAIN ONLINE PERMANENTLY.

USERS ARE ENCOURAGED TO EXAMINE THEIR JOBS CAREFULLY AND TO REPORT A5Y UNUSUAL DIFFICULTIES WHICH MAY BE A RESULT OF THE NEW RELEASE. COMPLETE DOCUMENTATION CONCERNING CHANGES AND ENHANCEMENTS WILL BE PUBLISHED AND RELEASED IN 2-4 WEEKS.
A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 17, REVISION 2

22 JULY 76

SUBJECT: ERROR NOTIFICATION - COMPUTATIONAL ITEMS IN COBOL68

IT WAS THOUGHT THAT ALL PROBLEMS WITH ALIGNMENT OF COMPUTATIONAL ITEMS WERE CORRECTED. IT HAS BEEN NOTED HOWEVER, THAT CERTAIN ARRANGEMENTS OF ITEMS STILL CAUSE INVALID ADDRESS TERMINATIONS DUE TO ALIGNMENT PROBLEMS.

IT IS SUGGESTED THAT PARTICULAR CARE BE TAKEN TO GROUP COMPUTATIONAL ITEMS TOGETHER WHEN AT ALL POSSIBLE. IN OTHER CASES, INSERT A SUFFICIENT AMOUNT OF FILLER WHERE NEEDED TO FORCE WORD ALIGNMENT. FOR OTHER INFORMATION, REFER TO THE ORIGINAL RELEASE OF THIS BULLETIN.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*  THIS SHOULD BE A PERMANENT ATTACHMENT TO THE ORIGINAL BULLETIN    *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 17, REVISION 1

11 JUNE 76

```
SUBJECT: REVISION NOTICE - COMPUTATIONAL ITEMS IN COBOL68
-------- -------- ------ - ------------- ----- -- -------
```

ALL KNOWN PROBLEMS WITH NON-WORD ALIGNED COMPUTATIONAL ITEMS HAVE BEEN CORRECTED. THE CORRECTIONS ARE EFFECTIVE IN COBOL68 LEVEL 16.0.13 AND ALL SUBSEQUENT LEVELS.

NOTE THAT WHILE WORD ALIGNED COMPUTATIONAL ITEMS ARE MORE EFFICIENT, IT IS NOT MANDATORY THAT COMPUTATIONAL ITEMS BE WORD ALIGNED.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*  THIS SHOULD BE A PERMANENT ATTACHMENT TO THE ORIGINAL BULLETIN    *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 17

07 OCTOBER 75

```
SUBJECT: ERROR NOTIFICATION - COMPUTATIONAL ITEMS IN COBOL68
-------- ----- ------------ - ------------- ----- -- -------
```

AS A GENERAL RULE, ALL COMPUTATIONAL ITEMS IN A COBOL68 PROGRAM MUST BEGIN AT A WORD BOUNDARY. THE PROGRAM-ER IS RESPONSIBLE FOR INSERTING AN APPROPRIATE AMOUNT OF FILLER TO ALIGN COMPUTATIONAL ITEMS TO THE PROPER BOUNDARY. AT THE PRESENT TIME, THE COMPILER WILL NOT AUTOMATICALLY DO THIS ALIGNMENT, AND FURTHERMORE, IT WILL NOT PRODUCE A SYNTAX ERROR OR WARNING MESSAGE WHEN COMPUTATIONAL ITEMS ARE -OT ALIGNED. AT EXECUTION TIME, HOWEVER, CERTAIN ACCESSES TO THESE MISALIGNED ITEMS WILL CAUSE THE PROGRAM TO BLOW UP WITH AN INVALID ADDRESS.

EXAMPLES

```
     INCORRECT                              CORRECT
     ---------                              -------
   01 STUFF.                             01 STUFF.
      03 A  PIC X(132).                     03 A  PIC X(132).
      03 B  PIC S9(11) COMP.                03 FILLER PIC X(4).
      03 C  PIC S9(11) COMP.                03 B  PIC S9(11) COMP.
                                            03 C  PIC S9(11) COMP.
```

```
                     PREFERRED
                     ---------
                  01 STUFF.
                     03 B  PIC S9(11) COMP.
                     03 C  PIC S9(11) COMP.
                     03 A  PIC X(132).
```

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 18

26 NOVEMBER 75

SUBJECT: CHANGES AND NEW FEATURES IN RELEASE 16.0 - BATCH MCP
-------- ------- --- --- -------- -- ------- ---- - ----- ---

THE ENTIRE MARK 16.0 RELEASE IS NOW ONLINE. THUS FAR, A MINIMUM NUMBER OF PROBLEMS HAVE BEEN EXPERIENCED. A COMPLETE ITEMIZATION OF CHANGES AND NEW FEATURES APPLICABLE TO THE BATCH MCP FOLLOWS.

A. THE LIBRARY MAINTENANCE ROUTINES HAVE BEEN COMPLETELY REWRITTEN TO INCORPORATE THE USE OF A NEW CONTROL CARD, THE "COPY" CONTROL CARD. THIS REWRITE ALSO EXTENDS THE USE OF "EXCEPT" LISTS TO THE "REMOVE" CONTROL CARD. DOCUMENTATION FOR THIS FEATURE IS LENGTHY AND CANNOT BE INCLUDED HERE. USERS DESIRING TO UTILIZE THE NEW LIBRARY MAINTENANCE SHOULD CONTACT THE

COMPUTING CENTER FOR DOCUMENTATION.

B. THE BATCH MCP IS NOW COMPATIBLE WITH THE TSSMCP IN THE AREA OF DISK FILE NAMING. THAT IS, IF THE FILES MFID IS ZERO THEN THE FILES FID IS MOVED TO THE MFID AND THE USERCODE IS MOVED TO THE FID. FOR EXAMPLE A FILE THAT PREVIOUSLY WOULD HAVE BEEN 0000000/FILNAME IS NOW FILNAME/USERCODE.

C. WHEN AN ERROR IS DETECTED IN A PACKET, THE PACKET WILL BE FLUSHED COMPLETELY (TO THE END CARD) - NOT JUST TO THE NEXT WAIT CARD.

D. THREE NEW TIME FUNCTIONS HAVE BEEN ADDED.

TIME(-3) - RETURNS CURRENT STATUS OF PACKETERR BIT
TIME(-4) - SETS THE PACKETERR BIT
TIME(-5) - RETURNS THE CURRENT VALUE OF PACKETACT

THESE THREE FUNCTIONS WERE ADDED TO GIVE A PROGRAM SOME CONTROL OVER THE PACKET. NOW A PROGRAM CAN "KILL" A PACKET BY A TIME FUNCTION (-4) RATHER THAN BY DS-ING ITSELF WITH A RUN TIME ERROR. ALSO, A PROGRAM CAN "SEE" IF A SISTER PROGRAM HAS RUN INTO TROUBLE (-3). AND FINALLY, A PROGRAM CAN TELL HOW MANY JOBS ARE RUNNING FROM THE PACKET AT THIS TIME (-5). NOTE THAT TIME(-4) WILL RETURN THE VALUE OF PACKETERR BEFORE IT SETS IT.

E. THE FILE CLOSE MESSAGES PLACED IN THE PACKET PAGE INFORMATION HAVE BEEN IMPROVED. PREVIOUSLY ONLY PRINTER-BACK-UP (PBD/PUD) CLOSE MESSAGES CONTAINED RECORD COUNTS. NOW ALL FILE CLOSE MESSAGES CONTAIN RECORD COUNTS. THE DISK FILE (DKA) CLOSE MESSAGES NOW INCLUDE THE CREATION DATE OF THE FILE, WHICH

PREVIOUSLY HAS NOT BEEN AVAILABLE. A NEW MESSAGE (RET) IS PRODUCED EACH TIME A FILE IS CLOSED BUT THE PROGRAM RETAINS CONTROL OF THE UNIT. THE MESSAGE IS IN THE SAME FORMAT AS THE OTHER MESSAGES (REL,LOK). PREVIOUSLY, THE ONLY MESSAGE PRODUCED FOR A RETAINED FILE WAS A REL MESSAGE AT THE END OF THE PROGRAM.

F. THE STACK TRACE HAS BEEN SHORTENED. NOW ONLY THE RCW"S ARE SHOWN ALONG WITH THE APPROPRIATE LOCATION INFORMATION A FULL STACK TRACE MAY BE OBTAINED IF THE PROGRAM IS EXECUTED WITH AN IO TIME LIMIT OF 1111.

G. THE SINGLE-QUOTE MAY NOW BE USED AS A BACKSPACE CHARACTER FOR ALL TYPE 19 FILES (REMOTE). THIS PROVIDES COMPATABILITY WITH BOTH CANDE AND R/C. NOTE HOWEVER THAT THE LESS-THAN CHARACTER MUST STILL BE USED AS THE BACKSPACE FOR ALL CONTROL CARDS AND MCP COMMUNICATION. THE SINGLE-QUOTE STILL *WILL NOT* WORK FOR ANY MCP COMMUNICATION.

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 19

26 NOVEMBER 75

SUBJECT: CHANGES AND NEW FEATURES IN RELEASE 16.0 - TIME-SHARING MCP

THE ENTIRE MARK 16.0 RELEASE IS NOW ONLINE. THUS FAR, A MINIMUM NUMBER OF PROBLEMS HAVE BEEN EXPERIENCED. A COMPLETE ITEMIZATION OF CHANGES AND NEW FEATURES APPLICABLE TO THE TIME-SHARING MCP FOLLOWS.

A. LIBRARY MAINTENANCE - SAME AS BATCH MCP DESCRIPTION A

B. TIME FUNCTIONS - SAME AS BATCH MCP DESCRIPTION D

C. FILE CLOSE MESSAGES - SAMF AS BATCH MCP DESCRIPTION E

D. SHORT STACK TRACE - SAME AS BATCH MCP DESCRIPTION F

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 20

26 NOVEMBER 75

SUBJECT: CHANGES AND NEW FEATURES IN RELEASE 16.0 - XALGOL

THE ENTIRE MARK 16.0 RELEASE IS NOW ONLINE. THUS FAR, A MINIMUM NUMBER OF PROBLEMS HAVE BEEN EXPERIENCED. A COMPLETE ITEMIZATION OF CHANGES AND NEW FEATURES APPLICABLE TO XALGOL FOLLOWS.

A. THE SEARCH STATEMENT WILL NOW RETURN THE ENTIRE DISK FILE HEADER IF THE SEARCH ARRAY IS GREATER THAN 40 WORDS. THE HEADER IS PLACED IN WORDS 10-39.

B. A NEW $-CARD - THE $ INCLUDE CARD

THE SYNTAX FOR THE $ INCLUDE CARD IS AS FOLLOWS:

```
  $ INCLUDE <COPY PART> <FILE PART> <SEQUENCE PART>

  <COPY PART> ::= <EMPTY> / + COPY
  <FILE PART> ::= <MULTI-FILE ID>/<FILE ID>   /
                  <MULTI-FILE ID>
  <MULTI-FILE ID> ::= <ALPHANUMERIC STRING
                         OF 7 OR FEWER CHARACTERS>
  <FILE ID> ::= <EMPTY> / <ALPHANUMERIC STRING>
  <SEQUENCE PART> ::= <STARTING SEQUENCE NUMBER> <ENDING SEQUENCE
                         NUMBER> / <&MPTY>
  <STARTING SEQUENCE NUMBER> ::= <UNSIGNED INTEGER>
  <ENDING SEQUENCF NUMBER> ::= <EMPTY> / - <UNSIGNED INTEGER>
```

SOME EXAMPLES ARE:

```
  $ INCLUDE A/B 1213-99932
  $ INCLUDE A 12321-77651
  $ INCLUDE+COPY SPECIAL/FILE 76333-124457
  $ INCLUDE A 12223
  $ INCLUDE A
  $ INCLUDE + COPY IT
```

INCLUDE INSTRUCTS THE COMPILER TO COMPILE THE SOURCE CODE ON THE DISK FILE <FILE PART> OVER THE RANGE <SEQUENCE PART> AS PART OF THE ENTIRE PROGRAM. IN THIS MANNER, THE USER CAN COMPILE ALL OR P1RT OF AN AUXILLARY FILE(S) INTO HIS PROGRAM. IF THE <FILE ID> IS NOT PRESENT, THE USERCODE IS USED AS THE <FILE ID>. THE STARTING AND ENDING SEQUENCE NUMBERS ARE

INCLUSIVE. 9F THE <SEQUENCE PART> IS EMPTY, THE ENTIRE FILE IS USED. IF ONLY THE STARTING SEQUENCE NUMBER IS PRESENT, THE FILE FROM THAT SEQUENCE NUMBER TO THE END OF THE FILE IS USED. IF BOTH SEQUENCE NUMBERS ARE PRESENT, THE FILE FROM THE STARTING SEQUENCE TO ENDING SEQUENCE, INCLUSIVE, IS USED. IF A NEW FILE IS BEING MADE, AND THE COPY PART IS EMPTY, ANY IMBEDDED $ INCLUDE CARDS WILL BE WRITTEN ON THE NEW FILE, BUT NOT THE INCLUDED FILES THEMSELVES. THIS PROVIDES THAT THE NEW FILE, WHEN IT ITSELF IS COMPILED, WILL INCLUDE THE FILES. WHILE AT THE SAME TIMF ALLOWING THE INCLUDED FILES TO BE UPDATED INDEPENDENTLY OF THE NEW FILE. IF A NEW FILE IS BEING MADE AND THE COPY PART IS PRESENT, THE IMBEDDED $ INCLUDE CARDS WILL NOT BE WRITTEN OUT ON THE NEW FILE, BUT RATHER THE INCLUDED RECORDS THEMSELVES WILL BE COPIED ONTO THE NEW FILE. THE COPY PART IS IGNORED IF A NEW FILE IS NOT BEING MADE. NOTE THAT INCLUDED FILES CAN HAVE $ INCLUDE CARDS IMBEDDED WITHIN THEM, AND THUS RECURSION ON THE $ INCLUDE CARDS CAN OCCUR.

C. THE GREATEST ALPHABETIC FORMAT HAS BEEN EXPANDED FROM A6 TO A7. THIS WILL ALLOW 7 ALPHA CHARACTERS TO BE STORED IN A WORD

WITHOUT HAVING TO USE PARTIAL WORD MOVES OR CONCATENATION.

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 21

26 NOVEMBER 75

SUBJECT: CHANGES AND NEW FEATURES IN RELEASE 16.0 - BASIC
-------- ------- --- --- -------- -- ------- ---- - -----

THE ENTIRE MARK 16.0 RELEASE IS NOW ONLINEC THUS FAR, A MINIMUM NUMBER OF PROBLEMS HAVE BEEN EXPERIENCED. A COMPLETE ITEMIZATI6N OF CHANGES AND NEW FEATURES APPLICABLE TO BASIC FOLLOWS.

A. BASIC PROGRAMS MAY NOW BE COMPILED FROM THE CARD READER FOR TERMINAL EXECUTION BY USING A

```
    ? BASIC COMMON=4
```

CONTROL CARD IN THE COMPILE-TO-LIBRARY DECK. PREVIOUSLY, THE INPUT AND PRINT STATEMENTS OF A PROGRAM COMPILED FROM THE CARD READER WOULD REFERENCE THE CARD READER AND PRINTER RESPECTIVELY. THIS NEW OPTION WILL ALLOW COMPILATION OF LARGE BASIC PROGRAMS FROM CARDS, BUT STILL ALLOW EXECUTION FROM TERMINALS WITHOUT THE BOTHER OF FILE EQUATION.

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 22

26 NOVEMBER 75

SUBJECT: CHANGES AND NEW FEATURES IN RELEASE 16.0 - COBOL68
-------- ------- --- --- -------- -- ------- ---- - -------

THE ENTIRE MARK 16.0 RELEASE IS NOW ONLINE. THUS FAR, A MINIMUM

NUMBER OF PROBLEMS HAVE BEEN EXPERIENCED. A COMPLETE ITEMIZATION OF CHANGES AND NEW FEATURES APPLICABLE TO COBOL68 FOLLOWS.

A. A NEW $-CARD OPTION - TSSCOPY

THE TSSCOPY $-CARD OPTION HAS BEEN IMPLEMENTED TO PROVIDE FOR LIBRARY FILES BEING IN EITHER BATCH OR CANDE FORMAT IS THE CASE FOR OTHER COMPILER INPUT. LIKE OTHER $-CARD OPTIONS, IT MAY BE SET, RESET, AND POPPED. THE ACTION TAKEN FOR BOTH SET AND RESET STATES IS DESCRIBED BELOW.

```
        TSSCOPY SET:
             THE INCOMING LIBRARY-FILE 9NPUT RECORDS ARE
             EXPECTED TO BE IN CANDE FORMAT:
                  COLUMNS  1 THRU 72 - COBOL STATEMENT
                  COLUMNS 73 THRU 80 - SEQUENCE NUMBER(ONLY 75
                               THRU 80 ARE SIGNIFICANT)
        TSSCOPY RESET:
             THE INCOMING LIBRARY-FILE INPUT RECORDS A9E
             EXPECTED TO BE IN STANDARD COBOL-STATEMENT
             FORMAT:
                  COLUMNS 1 THRU 6 - SEQUENCE NUMBER
                  COLUMNS 8 THRU 72 - COBOL STATEMENT
                  COLUMNS 73 THRU 80 - IDENTIFICATION, COMMENT,
                                       OR BLANK
        BY DEFAULT, TSSCOPY IS SET WHEN COMPILING
        FROM REMOTE THRU CANDE. IT IS RESET
        WHEN COMPILING IN A BATCH MODE FROM THE
        CARD READER.
```

PRIOR TO THIS PATCH, COPIED FILES WERE EXPECTED TO BE IN THE SAME FORMAT AS THE PRIMARY INPUT FILE (CARD). CONSEQUENTLY, BATCH COMPILATIONS WERE UNABLE TO COPY CANDE FORMAT LIBRARY FILE AND REMOTE COMPILATIONS WERE UNABLE TO COPY STANDARD COBOL-STATEMENT FORMAT LIBRARY FILES. THIS PATCH ALSO CORRECTS 1 PROBLEM WITH THE SEQUENCE-RANGE OPTION OF THE COPY STATEMENT. PRIOR TO THIS PATCH, THE FIRST SIX COLUMNS OF THE INCOMING LIBRARY-FILE RECORD WERE ALWAYS USED FOR THE SEQUENCE- RANGE

COMPARISON---EVEN IF THE INCOMING LIBRARY-FILE RECORD WAS IN CANDE FORMAT.

B. THE DISPLAY VERB HAS BEEN MODIFIED SO THAT THE MESSAGE WILL BE DISPLAYED AT THE CONSOLE ONLY IF THE UPON CONSOLF (OR SIMILAR PHRASE) PHRASE IS PRESENT. THE MESSAGE WILL APPEAR IN THE PACKET PAGE INFORMATION FOR ALL CASES, HOWEVER. THIS FEATURE WILL ALLOW THE DISPLAY VERB TO BE USED AS A DEBUGGING TOOL WITHOUT DELUGING THE OPERATOR WITH MESSAGES HE DOESN"T CARE ABOUT.

C. A NEW $-CARD OPTION, EBCDIC, HAS BEEN IMPLEMENTED. THE OPERATION OF AND CHARACTER SET PROVIDED BY THIS OPTION ARE BOTH IDENTICAL TO THAT FOR BASIC - REFER TO BULLETIN # 12.

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 23

26 NOVEMBER 75

SUBJECT: CHANGES AND NEW FEATURES IN RELEASE 16.0 - FORTRAN

THE ENTIRE MARK 16.0 RELEASE IS NOW ONLINE. THUS FAR, A MINIMUM NUMBER OF PROBLEMS HAVE BEEN EXPERIENCED. A COMPLETE ITEMIZATION OF CHANGES AND NEW FEATURES APPLICABLE TO FORTRAN FOLLOWS.

A. BOTH FORMS OF THE READ STATEMENT - READ 10,X AND READ(5,10) X - NOW ASK FOR "FILE5", THUS ALLOWING THE FORMS TO BE INTERMIXED THROUGHOUTA PROGRAM. ALSO, BOTH THE PRINT AND WRITE(6,...) STATEMENTS WILL WRITE ON ONE PRINTER FILE LABELED "FILE6".

B. A RANDOM NUMBER GENERATOR HAS BEEN IMPLEMENTED AS AN ADDITIONAL INTRINSIC FUNCTION FOR FORTRAN. THE FUNCTION "RND" GENERATES A SEQUENCE OF RANDOM NUMBERS BETWEEN ZERO AND ONE. EACH TIME IT IS CALLED WITH A ZERO ARGUMENT, IT PRODUCES THE NEXT NUMBER IN SEQUENCE IF THE ARGUMENT IS NEGATIVE, A RANDOM NUMBER IS GENERATED AND IS USED TO INITIALIZE THE SEQUENCE. IF A POSITIVE ARGUMENT IS USED, THE VALUE OF THE ARGUEMNT IS RETURNED AS THE RANDOM NUMBER AND IS USED TO INITIALIZE THE SEQUENCE. ARGUMENTS GREATER THAN 1 ARE NOT ALLOWED. IF THE FIRST CALL TO RND HAS A ZERO ARGUMENT, THE COMPUTER SUPPLIES A BUILT-IN NUMBER AS THE FIRST RANDOM NUMBER IN THE SEQUENCE.

```
   EXAMPLES:  X=RND(0)
              X=RND(ARG)
```

C. FREE FIELD IO STATEMENTS SUCH AS

```
              READ,I,J
              PRINT,A(K)
              PRINT,(A(J),J=1,10)
```

ARE NOW ALLOWED, THUS PROVIDING COMPATIBILITY WITH WATFIV.

D. EBCDIC TO BCL TRANSLATION MAY NOW BE INVOKED BY USE OF THE EBCDIC OPTION AS WELL AS THE HOL OPTION. BOTH OPTIONS PRODUCE IDENTICAL RESULTS. THE EBCDIC OPTION WAS ADDED TO PROVIDE COMPATIBILITY WITH NAMING CONVENTIONS USED IN BASIC AND COBOL68.

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 24

30 DECEMBER 75

SUBJECT: DOCUMENTATION FOR NEW AND IMPROVED FORTRAN FEATURES

SEVERAL ADDITIONS AND ENHANCEMENTS TO FORTRAN HAVE OCCURRED SINCE THE CURRENT MANUAL WAS RELEASED. A COMPLETELY NEW FORTRAN MANUAL HAS GONE TO THE PRESSES, BUT APPARENTLY IS BEING DELAYED FOR SOME REASON. DOCUMENTATION FOR SOME OF THE MORE INTERESTING IMPROVEMENTS IS BEING PUBLISHED HERE FOR THE BENEFIT OF OUR USERS. ALL FEATURES DESCRIBED ARE CURRENTLY OPERATIONAL. THE DOCUMENTATION IS TAKEN FROM THE PRE-RELEASE OF THE NEW FORTRAN MANUAL.

THE SUBJECTS FOR WHICH DOCUMENTATION IS ATTACHED ARE:

I. THE IMPLICIT STATEMENT

II. FORMAT SPECIFICATIONS AND STATEMENTS

III. FORMAT ERROR MESSAGES

IMPLICIT STATEMENT

THE NONEXECUTABLE IMPLICIT STATEMENT IS AN AUXILIARY TYPE STATEMENT WHICH ALTERS OR CONFIRMS THE DEFAULT TYPES ASSIGNED TO VARIABLES DUE TO THEIR INITIAL CHARACTERS.

THE GENERAL FORM IS:

```
IMPLICIT T(C) ...
```

WHERE T IS A TYPE NAME (REFER TO TYPE STATEMENTS), C IS AN INITIAL CHARACTER LIST, AND THE ELLIPSIS (...) INDICATES AS MANY REPETITIONS OF THE FORM T(C) AS DESIRED, WITH A COMMA PRECEDING EACH REPETITION.

SEVERAL IMPLICIT STATEMENTS ARE ALLOWED IN A PROGRAM UNIT, AND ANY PROGRAM UNIT MAY CONTAIN AN IMPLICIT STATEMENT. IF USED, THE IMPLICIT STATEMENT MUST BE THE FIRST STATEMENT OF THE MAIN PROGRAM OR THE SECOND STATEMENT OF A SUBPROGRAM. THE IMPLICIT STATEMENT APPLIES TO SYMBOLIC NAMES ONLY IN THE PROGRAM UNIT IN WHICH THE STATEMENT APPEARS, INCLUDING FUNCTION AND ENTRY NAMES AND DUMMY ARGUMENTS.

INITIAL CHARACTER LISTS.

EACH INITIAL CHARACTER LIST IS COMPOSED OF LIST ELEMENTS SEPARATED &Y COMMAS. EACH LIST ELEMENT MAY BE EITHER A SINGLE LETTER OR A RANGE OF LETTERS IN ALPHABETICAL ORDER. A RANGE IS DENOTED BY THE FIRST AND LAST LETTERS OF THE RANGE SEPARATED BY A MINUS SIGN (HYPHEN). THE IMPLICIT STATEMENT SPECIFIES A TYPE FOR ALL SYMBOLIC NAMES IN THE PROGRAM UNIT THAT BEGIN WITH ANY LETTER THAT APPEARS IN THE SPECIFICATION EITHER AS A SINGLE LETTER OR INCL4DED IN A RANGE OF LETTERS.

TYPE SPECIFICATION BY AN IMPLICIT STATEMENT MAY BE OVERRIDD5N BY EXPLICIT TYPE SPECIFICATION IN A TYPE STATEMENT OR IN A FUNCTION STATEMENT.

THE FOLLOWING ARE EXAMPLES OF IMPLICIT STATEMENTS:
(NOTE:THE FIRST TWO STATEMENTS HAVE IDENTICAL EFFECT.)

```
    IMPLICIT REAL(I-N)
    IMPLICIT REAL(A-Z)
    IMPLICIT DOUBLE PRECISION(D)
    IMPLICIT LOGICAL(A-C,L), REAL*8(D-F), COMPLEX(X)
```

FORMAT SPECIFICATIONS AND STATEMENTS

THE C FORMAT SPECIFICATION.

THE ALPHANUMERIC FORMAT SPECIFICATION CW CAUSES DATA TO BE TRANSFERRED TO OR FROM INTERNAL STORAGE AS CHARACTERS.

INPUT USING CW.

ON INPUT, THE CW SPFCIFICATION CAUSES THE CHARACTER STRING OF WIDTH W IN THE EXTERNAL FIELD TO BE ASSIGNED TO THE CORRESPONDING SIMPLE VARIABLE OR ARRAY ELEMENT IN THE I/O LIST. CHARACTERS ARE STORED INTERNALLY IN 6-BIT FORM, SIX CHARACTERS PER DATA WORD. THE CW FORMAT SPECIFICATION ALLOWS CHARACTERS TO BE PLACED RIGHT-JUSTIFIED INTO A DATA WORD WITH LEADING ZEROS. IF THE FIELD WIDTH, W EXCEEDS 6, THE RIGHT-MOST SIX CHARACTERS IN TH5 INPUT FIFLD ARE

STORED INTO THE CORRESPONDING VARIABLE IN THE I/O LIST AND THE REMAINING CHARACTERS ARE IGNORED.

OUTPUT USING CW.

ON OUTPUT, THE CW SPECIFICATION CAUSES THE CHARACTERS CONTAINED IN THE APPROPRIATE VARIABLE IN THE I/O LIST TO BE PLACED INTO AN EXTERNAL STRING OF LENGTH W. IF W IS GREATER THAN 6, THE SIX CHARACTERS CONTAINED IN THE DATA WORD WOULD BE PLACED RIGHT-JUSTIFIED INTO THE FIELD, PRECEDED BY BLANKS. IF W IS LESS THAN OR EQUAL TO 6, THE RIGHT-MOST W CHARACTERS IN THE DATA WORD WOULD BE PLACED INTO THE FIELD.

THE J FORMAT SPECIFICATION

THE INTEGER FORMAT SPECIFICATION JW CAUSES THE EXTERNAL CHARACTER STRING OF MOST W CHARACTERS TO BE ASSOCIATED WITH THE CORRESPONDING I/O LIST ELEMENT FOR PURPOSES OF DATA TRANSFER.

INPUT USING JW.

ON INPUT, THE JW SPECIFICATION FUNCTIONS IDENTICALLY TO THE IW SPECIFICATION.

OUTPUT USING JW.

ON OUTPUT, THE JW SPECIFICATION CAUSES THE VALUE OF THE CORRESPONDING I/O LIST ELEMENT TO BE PRINTED AS AN INTEGER CONSTANT IN THE MINIMUM FIELD NECESSARY TO CONTAIN THE VALUE WITHOUT EXCEEDING W. THE PLUS SIGN IS NOT PRINTED FOR POSITIVE QUANTITIES. IF THE VALUE TO BE PRINTED REQUIRES MORE THAN W CHARACTERS, W ASTERISKS ARE PRINTED. FLOATING POINT VALUES ARE ROUNDED TO AN INTEGER VALUE BEFORE PRINTING.

EXAMPLES:

INTERNAL VALUE		SPECIFICATION		EXTERNAL STRING
+23	->	J5	->	23
-23	->	J5	->	-23
+233	->	J3	->	233
-233	->	J3	->	***
0	->	J3	->	0

FORMAT SPECIFICATION MODIFIERS.

THE FORMAT SPECIFICATION MODIFIERS K AND $ ARE PROVIDED TO ALLOW INSERTION OF THE SPECIAL CHARACTERS COMMA (,) AND DOLLAR SIGN ($) IN AN OUTPUT FIELD. THEY MAY NOT BE USED WITH FORMAT

SPECIFICATIONS IN ARRAYS.

THE DOLLAR SIGN MODIFIER ($).

IF A $ PRECEDES AN IW, JW, GW.D, FW.D, EW.D, OR DW.D FORMAT PHRASE, A FLOATING DOLLAR SIGN IS INSERTED IN THE OUTPUT FIELD ADJACENT TO THE FIRST CHARACTER OF THE EDITED NUMBER. THE $ PRECEDING THE PHRASE MUST BE AFFIXED DIRECTLY TO THE PHRASE (E. G., $I10, $E20.10, OR 3P$F10.2). THE FLOATING DOLLAR SIGN CONSUMES ONE CHARACTER POSITION OF THE OVERALL FIELD WIDTH, W.

????????

THE K MODIFIER.

IF A K PRECEDES AN IW, JW, OR FW.D FORMAT PHRASE, COMMAS ARE INSERTED IN THE OUTPUT FIELD TO DELIMIT SUCCESSIVE GROUPS OF THREE DIGITS (ONLY TO THE LEFT OF THE DECIMAL POINT IN THE CASE OF THE FW.D FORMAT PHRASE).

THE K IS AFFIXED DIRECTLY TO THE PHRASE AND MAY BE INTERCHANGED WITH THE $ (E.G., KI10, $KJ4, OR K$F7.2). ON INPUT, THE COMMAS ARE IGNORED. EACH COMMA CONSUMES ONE CHARACTER POSITION OF THE OVERALL FIELD WIDTH, W.

THE ASTERISK TERMINATOR (*).

WHEN EDITING UNDER AN IW, JW, FW.D, EW.D, GW.D, OR DW.D FORMAT PHRASE, FORTRAN TERMINATFS EDITING UPON ENCOUNTERING AN ASTERISK IN THE DATUM, AND FLUSHES CHARACTERS UNTIL THE END OF THE FIELD WIDTH CONTAINING THE ASTERISKED DATUM. NOTE THAT THE ASTERISK TERMINATOR MAY NOT BE USED WITH FORMAT SPFCIFICATIONS IN ARRAYS.

A-A-A

FOR EXAMPLE, EDITING UNDER THE PHRASE 2I10, WITH DATA 742*BBBBBB17*BBBBBBB, THE VALUES 742 AND 17 ARE INPUT; SIMILARLY, WITH DATA 742BBBBBBB17BBBBBBBB, THE VALUES 7420000000 AND 1700000000 ARE INPUT.

THIS (*) CONSTRUCT ALLOWS COMMENTS WITHIN THE FIELD WIDTH ITSELF (E.G., 742*=SIZEB17*=TEMPER). ALSO, THE * IS USEFUL IN TERMINATING EXPONENTS, E.G., EDITING UNDER AN E20.10 PHRASE, WITH DATA 1.23E-17*BBBBBBBBBBB, THE VALUE 1.23E-17 IS INPUT, WHEREAS WITHOUT THE * THE EXPONENT MUST BE RIGHT-JUSTIFIED. FINALLY, IF THE FIRST NONBLANK, NONSIGN, NONZERO CHARACTER IS AN *, THE VALUE OF THE CORRESPONDING LIST ELEMENT REMAINS UNCHANGED. THIS USAGE ALLOWS THE ASTERISKED OUTPUT WHICH OCCURS DURING FIELD OVERFLOW TO BF SUITABLE INPUT.

THE ASTERISK FIELD WIDTH SPECIFIER (*).

IF AN ASTERISK (*) APPEARS IN A FORMAT SPECIFICATION LIST IN PLACE OF A FIELD WIDTH (W), A DECIMAL PLACE COUNT (D), SCALE FACTOR (S), OR A TAB POSITION (T), THE I/O LIST IS ACCESSED ONCE AND THE VALUE OF THE I/O LIST ELEMENT OBTAINED IS USED TO REPLACE THE *. A NEW I/O LIST ELEMENT IS REQUIRED FACH TIME AN * IS ENCOUNTERED IN THE SPECIFICATION LIST.

FOR EXAMPLE, THIS PROGRAM:

```
   10  FORMAT (I3,3X,I*)
       DO 5 I=0,7
       J=I+1
       K=10**I
    5  PRINT 10,I,J,K
       STOP
       END
```

PRODUCES THIS OUTPUT:

```
   0   1
   1   10
   2   100
   3   1000
   4   10000
   5   100000
   6   1000000
   7   10000000
```

IN THIS PROGRAM, THE VALUE OF J IN THE I/O LIST IS USED TO REPLACE THE ASTERISK IN THE SPECIFICATION LIST.

THE V FIELD SPECIFIER.

IF A V APPEARS IN A FORMAT SPECIFICATION LIST IN PLACE OF THE LETTER A, D, E, F, G, I, J, L, OR O IDENTIFYING A FORMAT SPECIFICATION, THE I/O LIST IS ACCESSED AND THE CONTENTS OF THE RIGHT-MOST SIX BITS OF THE DATA WORD REFERENCED ARF TREATED AS A CHARACTER TO BF USED IN THE ACTUAL FORMAT SPECIFICATION. THIS CHARACTER MUST BE ONE OF THE LETTERS JUST LISTFD; V IS NOT ACCEPTABLE. A NEW I/O LIST ELEMENT IS REQUIRED EACH TIME THE V IS ENCOUNTERED IN THE SPECIFICATION LIST.

THE FOLLOWING SAMPLE PROGRAM

```
      DATA B/"TEXT"/
      A="00000A"
      PRINT 1,A,B
    1 FORMAT(1X,V4)
      STOP
      END
```

PRODUCES THIS OUTPUT:

```
  TEXT
```

COMBINED EXAMPLE

THE FOLLOWING SAMPLE PROGRAM IS MORE COMPLEX IN NATURE AND DEMONSTRATES BOTH THE *-SPECIFIER AND THE V-SPECIFIER.

```
 10    FORMAT(2V10.2,*(*V*.*))
       F="00000F"; I="00000I"; A="00000A"; BIG=12345.67; TWO=2;
       SIX=6; ONE=1
       PRINT 10,I,SIX,BIG,TWO,ONE,F,SIX,TWO,SIX,TWO,A,SIX,BIG,I,F
       STOP; END
```

THIS SAMPLE PROGRAM WOULD PRODUCE THE FOLLOWING OUTPUT (B IS THE BLANK CHARACTER):

```
      BBBBBBB6BBBBB12346BB6.0000000I00000F
A-A-A
```

FORMAT ERROR MESSAGES

SEVERAL ERROR MESSAGES ARE PROVIDED TO INFORM THE USER WHEN FORMATTING ERRORS ARE MADE.

INPUT DATA ERROR MESSAGE.

WHENEVER ERRONEOUS DATA IS ENCOUNTERED OR THE TYPE DOES NOT MATCH THE FORMAT, AN ATTEMPT IS MADE TO BRANCH TO THE ERR ACTION LABEL (THIS IS ONLY FOR FORMATS NOT CONTAINED IN AN ARRAY). IF THERE IS NO ERR ACTION LABEL, THE OBJECT PROGRAM IS DS-ED FOR ERRONEOUS DATA WITH ONE OF THE FOLLOWING MESSAGES.

```
      "DATA ERR #N:  FMT IS FP, TYP=TY, COL#CL, CHR=CR, REC#RN:
      <FOLLOWED BY THE USUAL PROGRAM, SEG, AND ADDR INFORMATION>"
   OR
      "DATA ERR #N:  VFP=FP, TYP=TY, COL#CL, CHR=CR, REC#RN:
      <FOLLOWED BY THE USUAL PROGRAM, SEG, AND ADDR INFORMATION>."
```

WHERE:

N IS ONE OF THE DIGITS 1, 2, OR 3: 1=ILLEGAL CHARACTER IN DATA

FIELD; 2=NONINTEGRAL DATUM READ UNDER AN I PHRASE; AND 3=NUMERIC OVERFLOW (>3777777777777777 FOR OCTAL, > 549755813887 FOR INTEGER AND LOGICAL, AND >4.31359146672*10**68 FOR ALL OTHERS);

FP IS THE FORMAT PHRASE EDITING THE ERRONEOUS DATUM, E.G., E10.2;

TY IS THE LIST ELEMENT TYPE (R, I, L, D, AND C FOR REAL, INTEGER, LOGICAL, DOUBLE PRECISION, AND COMPLEX RESPECTIVELY); AND

CL IS THE COLUMN OR CHARACTER NUMBER WITHIN RECORD NUMBER RN WHERE THE ERRONEOUS CHARACTER CR IS FOUND (RECORD NUMBERING STARTS AT 1).

THE FIRST FORM IS USED WHEN THE FORMAT PHRASE EDITING THE ERRONEOUS DATUM HAS NO VARIABLE OR DYNAMIC PARTS, AND THE SECOND FORM IS USED WHEN THERE ARE VARIABLE OR DYNAMIC PARTS.

IN THE SECOND FORM, VFP IS THE VARIABLE FORMAT PHRASE AS COMPILED (E.G., -*P, 2G*.6, OR V10), AND FP IS THE FORMAT PHRASE AFTER THE VARIOUS DYNAMIC PARTS ARE EVALUATED (AT RUN-TIME). FOR EXAMPLE, VFP=FP MIGHT READ 2F7.*=2F7.2, OR V10=I10, OR *V*.*=I14, ETC.

TWO COMPLETE SAMPLE ERROR MESSAGES ARE:

```
        "DATA ERR #1:  FMT IS 2F10.6, TYPE=R, COL#62, CHR=A, REC#626:
        <FOLLOWED BY THE USUAL PROGRAM, SEG, AND ADDR INFORMATION>"
    AND
        "DATA ERR #2:  *V10=7I10, TYP=I, COL#12, CHR=., REC#217:
        <FOLLOWED BY THE USUAL PROGRAM, SEG, AND ADDR INFORMATION>."
```

A-A-A

VARIABLE FORMAT ERROR MESSAGE.

```
        "VARBL FMT ERR:  VFP=FP:  <USUAL PROGRAM, SEG, AND ADDR
        INFORMATION>."
```

WHERE VFP IS THE FORMAT PHRASE WITH ITS VARIOUS DYNAMIC PARTS (THE PHRASE AS IT APPEARED AT COMPILE TIME), FOR EXAMPLE:

```
        -*P, 2G*.6, OR 7V10.4
```

AND FP IS THE SAME PHRASE AFTER AN ATTEMPT HAS BEEN MADE TO EVALUATE THE VARIOUS DYNAMIC PARTS. THE FAILURE OF THE ATTEMPT IS REVEALED IN FP ITSELF. FOR EXAMPLE, VFP=FP READS AS ANY OF THE FOLLOWING:

```
        *V*=<7264.12>V*
        *V*=1<-6>*
        *V*=2E12.<MISSING D>
        *V=7I<MISSING W>
        4V=4E<MISSING W AND D>
```

A COMPLETE SAMPLE ERROR MESSAGE IS:

"VARBL FMT ERR: *E16.=6E16.<-8.2E14>: <FOLLOWED BY THE USUAL PROGRAM, SEG, AND ADDR INFORMATION>.

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 25, REVISION 1

11 JUNE 76

SUBJECT: REVISION NOTICE - SUBSCRIPTING IN COBOL68

SEVERAL PATCHES CONCERNING SUBSCRIPTING IN COBOL68 HAVE BEEN INSTALLED DURING THE PAST MONTH. ALL KNOWN PROBLEMS IN THIS AREA HAVE BEEN CORRECTED. THESE CORRECTIONS ARE EFFECTIVE IN COBOL68 LEVEL 16.0.15 AND ALL SUBSEQUENT LEVELS.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*  THIS SHOULD BE A PERMANENT ATTACHMENT TO THE ORIGINAL BULLETIN    *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 25

14 JANUARY 76

SUBJECT: SUBSCRIPTING IN COBOL68

USING A FORMULA FOR A SUBSCRIPT WILL CAUSE INCORRECT RESULTS TO BE PRODUCED. THE SUBSCRIPT FORMULA IS NOT EVALUATED CORRECTLY AND CAUSES THE OFSET FOR REFERENCING THE ARRAY TO BE INCORRECT.

EXAMPLES:

```
          IF AR(NDS + 6) EQUAL SOMETHING MOVE 1 TO SOMETHING-ELSE.
          MOVE AR(NDX / 6 + 2) TO LINE-P.
```

UNTIL A CORRECTION CAN BE INSTALLED, IT IS RECOMMENDED THAT ALL SUBSCRIPT COMPUTATION BE DONE SEPARATELY.

EXAMPLES:

```
          COMPUTE TMP = NDX / 6 + 2.
          MOVE AR(TMP) TO LINE-P.
```

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN #26

06 APRIL 76

SUBJECT: GUIDELINES FOR TAPE AND DISK FILE ATTRIBUTES

A. FOR MAXIMUM EFFICIENCY AND TO AVOID THE LOSS OF RECORDS THE FOLLOWING RULES SHOULD BE FOLLOWED WHEN DESIGNING TAPE AND DISK FILES. ANY DEVIATION FROM THESE RULES MAY CAUSE RECORDS TO BE LOST OR INACCESSIBLE.

GENERAL

1. RECORD SIZE SHOULD ALWAYS BE A MULTIPLE OF 8 (CHARACTERS).
2. SUGGESTED BLOCK SIZE RANGE IS 2400 - 3600 (CHARACTERS).
3. FILES SHOULD BE WRITTEN AND READ WITH THE SAME RECORD SIZE AND BLOCKING FACTOR.

ADDITIONAL RULES FOR DISK FILES

1. BLOCK SIZE (IN CHARACTERS) SHOULD BE A MULTIPLE OF 240 (CHARACTERS).
2. PAGE SIZE (IN RECORDS) SHOULD BE A MULTIPLE OF THE BLOCKING FACTOR (IN RECORDS). [NOTE: IN FORTRAN PAGE SIZE = AREA/20]

B. RECORDING MODE SHOULD BE STANDARD (BINARY) FOR ALL DISK FILES AND FOR ALL TAPE FILES WHICH ARE USED LOCALLY. NON-STANDARD (ALPHA) TAPES SHOULD BE WRITTEN ONLY WHEN THE TAPE IS TO BE SENT TO A NON-B5500 SITE.

C. STANDARD LABELS ARE HIGHLY RECOMMENDED. IT IS ALSO RECOMMENDED THAT THE FILE NAME BE A TWO-PART ID (MFID/FID) TO MAKE THE NAME MORE UNIQUE.

D. MULTI-FILE TAPES ARE NOT GENERALLY RECOMMENDED FOR DATA TAPES WHICH ARE FREQUENTLY UPDATED, BUT THEY SOMETIMES PROVIDE A HANDY MEANS FOR COMPACT BACKUP STORAGE.

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 27

08 APRIL 76

SUBJECT: NOTES ON FILE EQUATION

1. THERE IS EVIDENCE INDICATING THAT THE FILE EQUATION CARD DOES NOT PROPERLY HANDLE THE RECORDING MODE OF A FILE (COBOL AND COBOL68 PROGRAMS ONLY). USERS SHOULD EXERCISE CARE WHEN EQUATING TO A FILE THAT HAS A DIFFERENT RECORDING MODE THAN THAT ORIGINALLY COMPILED INTO THE PROGRAM. ALSO, THE RECORD AND BLOCK SIZES OF THIS FILE (TO WHICH YOU HAVE EQUATED) SHOULD BE THE SAME AS THOSF COMPILED IN THE PROGRAM SINCE FILE EQUATION DOES NOT HANDLE THESE ATTRIBUTES.

EXAMPLE:

A PROGRAM USING A FILE DECLARED AS A CARD FILE (ALPHA RECORDING MODE) BUT EQUATED TO A STANDARD TAPE FILE (BINARY RECORDING MODE) MAY CAUSE SEVERE AND UNEXPECTED PROBLEMS.

2. PLEASE NOTE THAT WHEN A FILE IS EQUATED TO DISK VIA THE FILE EQUATION CARD THE OUTPUT MEDIA SPECIFICATION "DISK" WILL CAUSE THE FILE TO BE A RANDOM ACCESS FILE WHEREAS THE OUTPUT MEDIA SPECIFICATION "SERIAL" WILL CAUSE THE FILE TO BE A SEQUENTIAL FILE.

3. FOR FILE EQUATION CARDS THE OUTPUT MEDIA "PRINTER" IS NOT A VALID UNIT. FOR EQUATION TO THE PRINTER THE MEDIA SHOULD BE "PRINT".

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 28

08 APRIL 76

SUBJECT: SEGMENTATION IN FORTRAN
-------- ------------ -- -------

CURRENTLY, THE DEFAULT SEGMENT SIZE LIMIT OF A FORTRAN PROGRAM SEGMENT IS 512 WORDS. THIS SIZE IS PROVEN TO BE THE MOST EFFICIENT FOR MOST PROGRAMS BUT MAY BE VARIED BY USE OF THE

```
                  $ SET SEGMAX [LIMIT]
```

CARD WHERE [LIMIT] IS THE SEGMENT LIMIT IN WORDS.

```
      EXAMPLE: $ SET SEGMAX 256
```

REGARDLESS OF THE SEGMENT SIZE LIMIT, THE COMPILER DOES NOT ALWAYS BREAK A SEGMENT IN THE LOGICAL PLACE. FOR EXAMPLE, A SEGMENT BREAK MAY OCCUR WITHIN A DO LOOP, THUS MEANING A CONSIDERABLE AMOUNT OF OVERLAYING IS DONE DURING EXECUTION OF THE DO LOOP. SINCE THE DO LOOP SPANS 2 SEGMENTS WHICH PROBABLY WILL NOT BE IN CORE AT THE SAME TIME, IO TIME FOR THE JOB WILL BE INCREASED. ONE OR TWO CASES OF THIS IN A PROGRAM MAY NOT CAUSE A LARGE PROBLEM, BUT IN SIZABLE PROGRAMS WHERE SEVERAL ILLOGICAL SPLITS USUALLY OCCUR A CONSIDERABLE PROBLEM MAY DEVELOP. FUTHERMORE, IN SOME PROGRAMS WITH LARGE WRITE STATEMENTS , A BREAK MAY OCCUR IN THE MIDDLE OF THE WRITE LIST. THIS MAY CAUSE THE WRITE STATEMENT TO LOOP INFINITELY WHILE PRINTING GARBAGE DATA.

THE ONLY SIMPLE SOLUTION FOR THE ABOVE TWO PROBLEMS IS THE INSERTION OF A

$ SET SEGMENT

CARD IN APPROPRIATE PLACES TO CAUSE FORCED SEGMENT BREAKS IN THE LOGICAL PLACES. REARRANGING THE ORDER OF THE CODE MAY ALSO BE HELPFUL. HAVING GROUPS OF INFREQUENTLY USED CODE GROUPED IN SEGMENTS BY THEMSELVES ALONG WITH FREQUENTLY USED SECTIONS IN OTHER SEGMENTS WILL ALSO IMPROVE EFFICIENCY. THE OVERALL OBJECTIVE IS TO HAVE CODE SEGMENTS GENERATED IN A MANNER THAT WILL CAUSE THE LEAST AMOUNT OF OVERLAYING.
A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 29

13 MAY 76

SUBJECT: REVISION NOTICE - COPY VERB IN COBOL68

ALL KNOWN PROBLEMS WITH THE "REPLACING" OPTION OF THE "COPY" VERB IN COBOL68 HAVE BEEN CORRECTED. THE CORRECTION IS EFFECTIVE IN COBOL68 LEVEL 16.0.14 AND ALL SUBSEQUENT LEVELS.
A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 30, REVISION 1

13 SEPTEMBER 76

SUBJECT: REVISION NOTICE -- COBOL68 SORT

ALL KNOWN PROBLEMS WITH USING/GIVING AND INPUT-PROCEDURE/GIVING SORTS HAVE BEEN CORRECTED. THE CORRECTION IS EFFECTIVE WITH THE COBOL68 COMPILER LEVEL 16.0.16.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*  THIS SHOULD BE A PERMANENT ATTACHMENT TO THE ORIGINAL BULLETIN    *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

A-A-A

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 30

02 JULY 76

SUBJECT: ERROR NOTIFICATION - COBOL68 SORT
-------- ----- ------------ - ------- ----

SOMETIME BACK A PROBLEM EXISTED IN COBOL68 WHEREIN A SYSTEM HANG WOULD RESULT IF BOTH THE USING AND GIVING PORTIONS OF THE SORT REFERENCED THE SAME FILE (EG, SAME FD). THAT PARTICULAR PROBLEM WAS CORRECTED ABOUT A YEAR AGO.

IT HAS COME TO OUR ATTENTION THAT A SIMILAR PROBLEM HAS DEVELOPED, HOWEVER. THE CURRENT PROBLEM IS THAT A PROGRAM HAVING ITS INPUT PROCEDURE REFERENCE THE SAME FILE AS A GIVING CLAUSE OF THAT SORT WILL SUFFER AN ABNORMAL TERMINATION. IT IS SUSPECTED THAT THIS MIGHT ALSO BE THE CASE FOR A USING/OUTPUT PROCEDURE COMBINATION.

A CORRECTION IS BEING PURSUED; HOWEVER, IT IS RECOMMENDED THAT AS A TEMPORARY MEASURE EACH SORT REFERENCE TWO SEPARATE FILES (NOTE THAT EACH FILE MAY HAVE THE SAME VALUE OF ID). THIS SUGGESTION SHOULD ALLEVIATE ANY UNEXPECTED PROBLEMS UNTIL A CORRECTION IS PROCURED.
A-A-A

ECU COMPUTING CENTER

13 JULY 76

SUBJECT: COBOL68 RESERVED WORDS

THE FOLLOWING LIST OF RESERVED WORDS ARE OMITTED FROM THE CURRENT COBOL68 REFERENCE MANUAL (#1081148 OF JANUARY, 1975). THESE WORDS SHOULD BE ADDED TO THE EXISTING LIST OF RESERVED WORDS IN THE MANUAL.

AS

CANCEL
CLEAR

LOCAL

MODS
MYUSE

NAME
NOTE
NOTES

PROTECT

RANK
REMARKS

SECONDS
SET
SP
STATE

TASK-ID
TASK-NAME

WHEN
WITH
WORDS
WORKING-STORAGE
WRITE

ZERO
ZEROES

A-A-A

ZEROS

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 32

16 JULY 76

SUBJECT: ERROR NOTIFICATION - XALGOL WRITE STATEMENT
-------- ----- ------------ - ------ ----- ---------

IT HAS BEEN DISCOVERED THAT IN RELATION TO A REMOTE FILE, THE CONSTRUCT

```
WRITE(UNIT[STOP], ... );
```

DOES NOT FUNCTION CORRECTLY WHEN THE PROGRAM CONTAINING IT IS RUN UNDER THE BATCH MCP. THE CONSTRUCT FUNCTIONS PROPERLY UNDER TIME SHARING.

THE PROBLEM IS THAT GARBAGE LEFT IN THE BUFFER BY THE PREVIOUS WRITE TO THE REMOTE FILE IS PRINTED REGARDLESS OF THE [STOP] OPTION. FOR EXAMPLE, THE STATEMENT

```
WRITE(REM[STOP],1,A[*]);
```

WOULD PRINT THE CONTENTS OF WORD 1 OF ARRAY A FOLLOWED BY WHATEVER GARBAGE IS CONTAINED IN THE REMAINDER OF THE 28-CHARACTER BUFFER. THE DESIRED EFFECT WAS TO HAVE A[1] WRITTEN AND STOP THE CARRIAGE IMMEDIATELY AFTER IT"S LAST CHARACTER, AND THIS IS PRECISELY THE RESULT WHEN THE PROGRAM IS RUN UNDER TIME SHARING.

NO TEMPORARY SOLUTION HAS BEEN FOUND WHICH WILL PRODUCE THE EXACT DESIRED RESULTS. A CORRECTION IS BEING PURSUED AND WILL BE INSTALLED AS SOON AS POSSIBLE.

ECU COMPUTING CENTER

B5500 SYSTEMS BULLETIN # 33

SEPTEMBER 22, 1976

SUBJECT: ERROR NOTIFICATION BASIC FILE HANDLING ERRORS

USE OF THE END-OF-FILE BRANCH IN BASIC CAN CAUSE SYSTEM HANGS AT EXECUTION TIME. BASIC USERS SHOULD BE AWARE THAT PROGRAMS CONTAINING THIS OPTION MAY CAUSE PROBLEMS.

THE FOLLOWING WILL CERTAINLY CAUSE AN ERROR IF DATAM CONTAINS LESS THAN 10 NUMBERS:

```
100 DIM X(10)
200 MAT INPUT FILE DATAM:300,X
300 MAT PRINT X
400 STOP
500 END
```

```
 LABEL  000000000PRNT   00177279?USER=SPO    ; EXECUTE COPY   /SPO                                  COPY   /SPO
```