Burroughs

B 6000 Series System Notes

MARK III.0 RELEASE

PRICED ITEM

Printed in U.S. America

July 1978

5010713

Correspondence regarding this document should be addressed directly to Burroughs Corporation, P.O. Box 4040, El Monte, California 91734, Attn: Publications Department, TIO–West.

TABLE OF CONTENTS

MARK 3.0

DOCUMENT CHANGES NOTES (D NOTES)

GENERAL

D2265 GENERAL - STANDARDIZATION OF COMPILER FILES

The blocksize and areasize attributes of all files created by the compilers have been changed to conform to the standard for all software.

D2507 GENERAL - FIRMWARE FILES AND LISTINGS

Four files dealing with diskpack firmware are in the II.9.2 and III.0 releases. SYSTEM/FIRMWARE/215 and SYSTEM/FIRMWARE/225AND235 are firmware files suitable for a host load on all B6800's and B6700's with Model III Multiplexors. SYSTEM/FIRMWARE/215 is release 2.0 Revision D for 215 diskpacks; SYSTEM/FIRMWARE/225AND235 is release 2.0 Revision D for 225 and 235 subsystems. SYMBOL/FIRMWARE/215 and SYMBOL/FIRMWARE/225AND235 are printer backup disk files of listings of the assembly of the corresponding firmware file. A listing of the 225AND235 firmware may be produced by copying SYMBOL/FIRMWARE/225AND235 to DISK or PACK and entering the following control card image:

```
?PB "SYMBOL/FIRMWARE/225AND235" SAVE.
```

GENERAL

P1414 GENERAL - COPYRIGHTS, VERSION LEVELS UPDATED

The copyright notices have been updated for 1978.

Version levels for all software products have been set to the current level.

ALGOL

D2241 ALGOL - "SEGDESCABOVE" $ OPTION

The following changes have been made in the implementation of the SEGDESCABOVE $ option:

1. The value is no longer optional; an explicit value must be given.

2. The option may now be used even after compilation has begun; in this case, the given value is added to the current size of the D1 stack.

3. The value may be a compile-time variable, as well as a constant.

D2256 ALGOL - "$WARNSUPR" IMPLEMENTED

The WARNSUPR dollar card option now causes the printing of warning messages to be suppressed. WARNSUPR is RESET by default.

D2283 ALGOL - UPDATE INFO LEVEL

The compiler's info level has been changed. To do a sepcomp or loadinfo, the info level of the host and the procedure to be bound must match exactly. The new info level is 30.

D2289 ALGOL - "OFFSET" FUNCTION

An integer function, OFFSET (<pointer-expression>), is now available. This function calculates the number of characters or words, in terms of the character or word size of the pointer expression, that the pointer expression is indexed from the front of an array row. The OFFSET function returns a total offset from the point of the first row for pointer expressions which point at segmented arrays.

D2290 ALGOL - DEFINE INVOCATION IN DECLARATIONS

Defines are not expanded in declarations if they occur where the identifier being declared is expected.

Example:

```
DEFINE P = PROCEDURE #;
REAL P;
```

The P in the second line of the above example will not be expanded to PROCEDURE. This is necessary so that define identifiers can be redeclared.

D2403 ALGOL - WARNINGS ON USE OF "BCL" CONSTRUCTS

Use of BCL constructs will now cause a warning to be emitted indicating that the program is not portable to EBCDIC machines. Only the first occurrence of BCL constructs in a program or (in the case of separately-compiled program units) program unit will cause a warning to be emitted. A warning message will also appear in the trailer information at the end of the listing.

D2427 ALGOL - STRING IMPLEMENTATION

String variables allow for the storage and manipulation of character strings within a program.

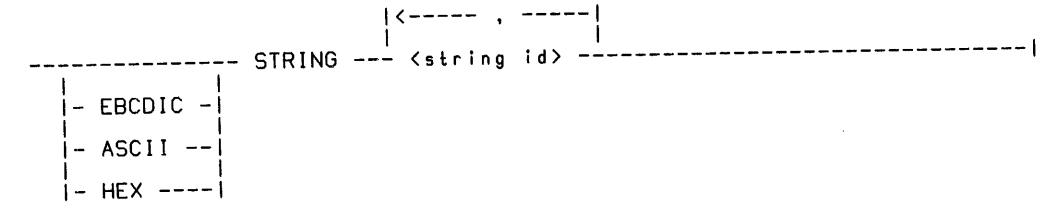
A string is a structured data type whose components are all of one character type. A string has two attributes: length and contents. No trailing blank or null characters are added to the string, so the actual length of the string is exactly the total number of characters stored into the string variable.

The syntactic items used in this description match those in the ALGOL Manual.

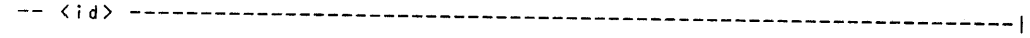
STRING DECLARATION

All string variables must be explicitly declared in a string declaration statement:

<string declaration>



<string id>



String variables have flexible storage areas whose working size is completely under control of the MCP.

The maximum string length allowed is 2 ** 16 - 2 characters.

All string variables in a <string declaration> are of the same character type. If no character type is specified, the default type will be used.

The default character type of EBCDIC may be changed by using a $ASCII or a $BCL card; however, the BCL type is illegal for string variables.

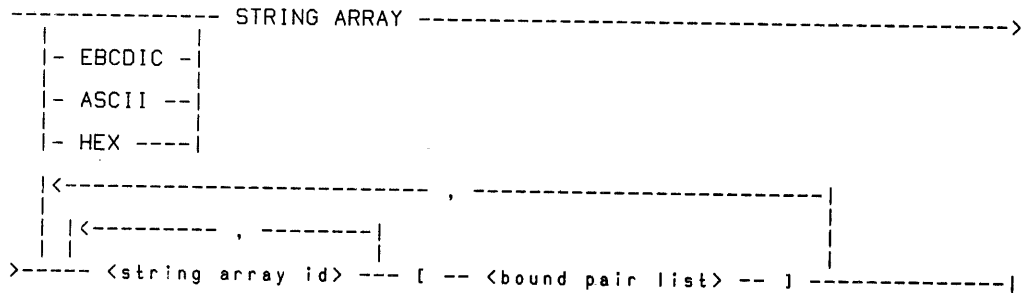
Examples

```
ASCII STRING S1, S2, S3;
HEX STRING S4;
EBCDIC STRING S5, S6, S7, S8;
STRING S9;
```

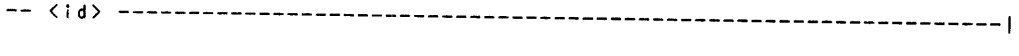
STRING ARRAY DECLARATION

A string array declaration can be used to declare a string array. A string array is an array of string variables.

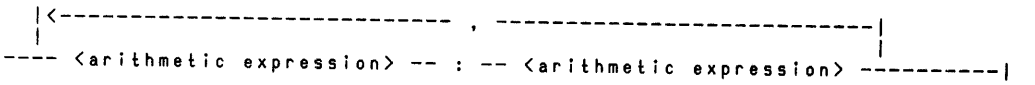
<string array declaration>



<string array id>



<bound pair list>



If no character type is specified in a string array declaration, the default character type will be used.

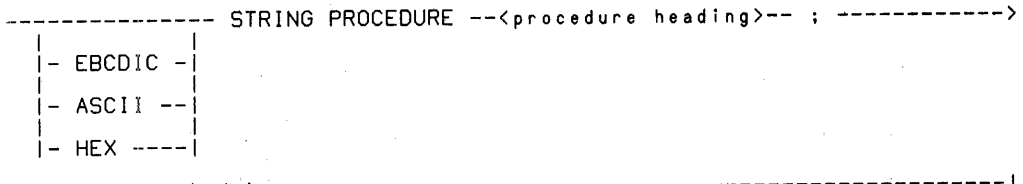
The <bound pair list> specification is exactly the same as for an ARRAY declaration described in the ALGOL Manual.

Examples

```
STRING ARRAY SA, SB, SC[0 : 10];
EBCDIC STRING ARRAY ESA[1 : 15], ESB, ESC[0 : 10, 0 : 10];
```

STRING PROCEDURE DECLARATION

The syntax for a procedure declaration is extended to allow a string-valued procedure.



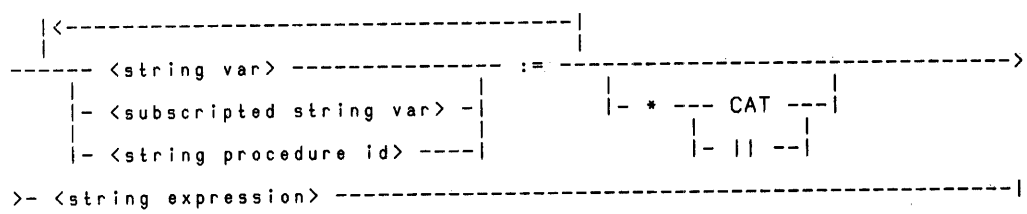
If no character type is specified in the string procedure declaration, the default character type will be assumed.

```
STRING PROCEDURE STR;...
HEX STRING PROCEDURE HEXPROC(R); REAL R;...
```

STRING ASSIGNMENT

A string variable is assigned a value via the <string assignment statement>:

<string assigrment statement>



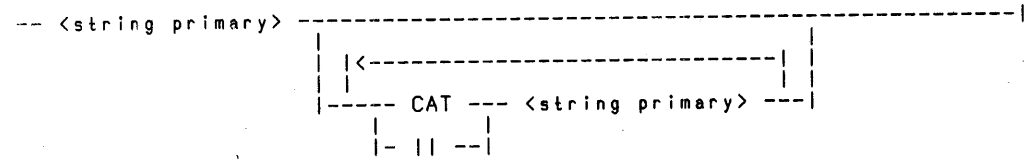
The result of the expression on the right hand side of the assignment operator must be a string of the same character type as the declared type of the string variable on the left hand side. If this condition is not satisfied, a syntax error will occur.

Note that embedded assignment is not allowed for strings. The following is syntactically illegal:

```
S1 := DROP(S2 := "ABC",2);
```

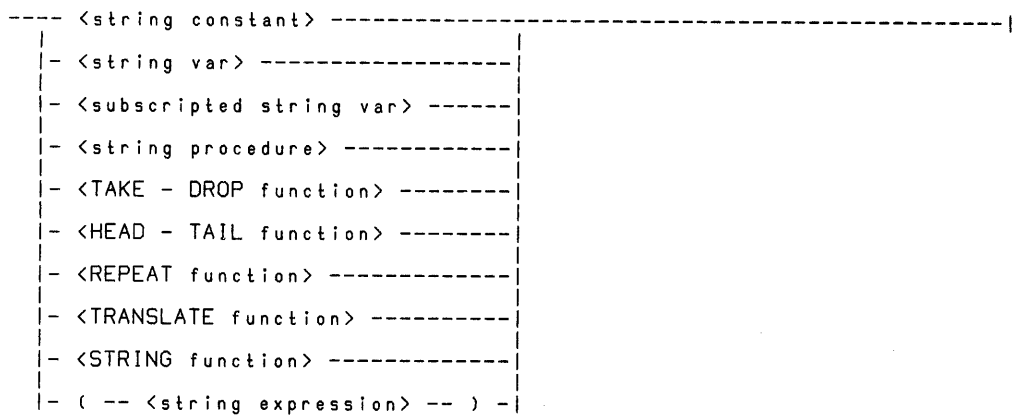
STRING EXPRESSIONS

<string expression>

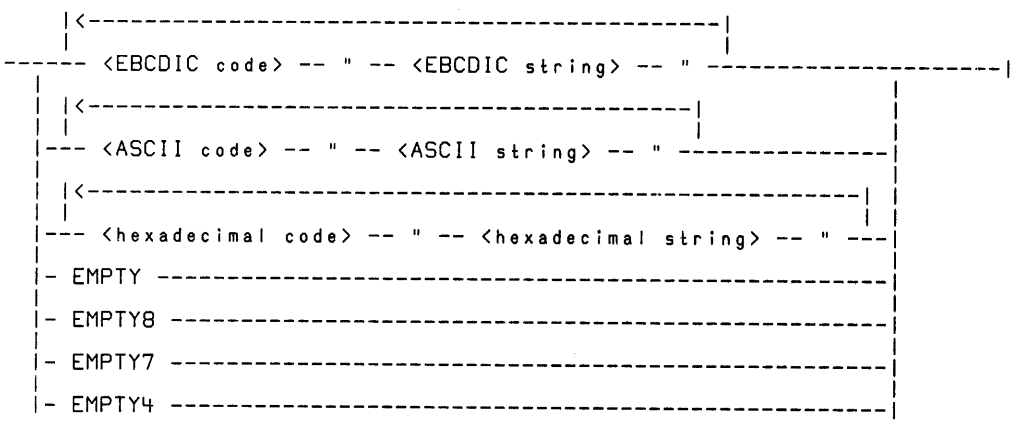


The allowable forms of <string primary> are given by:

<string primary>



<string constant>



A string constant can be composed of EBCDIC(8-bit), ASCII(7-bit in 8-bit format), and hexadecimal(4-bit) characters.

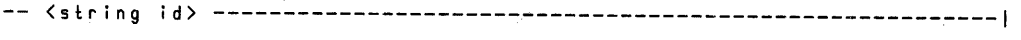
The string code determines the interpretation of the characters between the quotes as described in the ALGOL Manual; however, it has no effect on the justification of a string. A string is always left justified.

The reserved words EMPTY8, EMPTY7 and EMPTY4 represent null strings of the character type EBCDIC, ASCII, and HEX, respectively. The reserved word EMPTY represents a null string of the default character type.

Examples

```
S := 8"ABCD123";                result = "ABCD123"
S := ""WHY"48"6F""";            result = ""WHY?""
S := EMPTY8;                    result = ""
```

<string var>

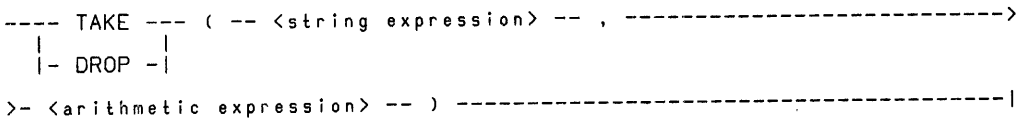


<string var> is a <string id> declared in a <string declaration>.

<subscripted string var>

THE TAKE - DROP FUNCTION

<TAKE - DROP function>



The TAKE function returns a new string whose value is a copy of the first <arithmetic expression> number of characters taken from the <string expression>. An error will occur if the value of <arithmetic expression> is greater than the number of characters in <string expression>, or <arithmetic expression> is less than zero. If <arithmetic expression> is zero, the result is the null string. If <arithmetic expression> is the length of the <string expression>, the result is the same as the value of <string expression>.

The result of the DROP function is a copy of the characters remaining in <string expression> with the first <arithmetic expression> number of characters removed. The value of <arithmetic expression> is limited as in the TAKE function. If the value of <arithmetic expression> is zero, the result is the same as the value of <string expression>. If the value of <arithmetic expression> is the length of the <string expression>, the result is the null string.

For any <string expression> S and any <arithmetic expression> A in the range 0 <= A <= LENGTH(S), the following relation is always TRUE:

```
S = TAKE(S, A) || DROP(S, A)
```

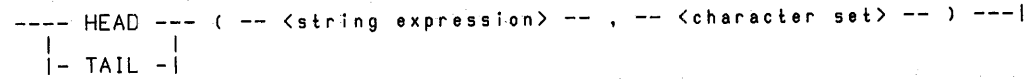
Examples

Assume string S has length of 6 and contains "ABCDEF":

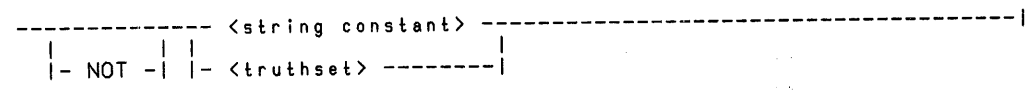
```
TAKE(S, 2)                    gives   "AB"
TAKE(S, 4)                    gives   "ABCD"
DROP(S, 2)                    gives   "CDEF"
DROP(TAKE(S, 4), 2)           gives   "CD"
TAKE(S, 8)                    gives   run-time error
DROP(S, - 2)                  gives   compile-time error
```

THE HEAD - TAIL FUNCTION

<HEAD - TAIL function>



<character set>



A <character set> is a collection of characters used to control the action of the HEAD and TAIL functions. NOT is used to indicate a character set consisting of all <character string code> characters except those specified. A <truthset> is defined in the ALGOL Manual. A <character set> must be of the same character type as the <string expression>; otherwise, a compile-time error will occur.

The HEAD function returns a new string consisting of a copy of all the leading characters in <string expression> that belong to the characters specified by <character set>. If the first character in <string expression> is not a member of the <character set>, a null string is returned.

The TAIL function returns a new string consisting of a copy of all the characters in <string expression> that remain after the removal of all the leading characters that belong to <character set>. If all characters in <string expression> are members of the specified <character set>, a null string is returned.

For any <string expression> S and any <character set> C, the following relation is always TRUE:

```
S = HEAD(S, C) || TAIL(S, C)
```

Examples

Let S be a string of length 9 and contain "ABC/1-2+3":

```
HEAD(S, NOT "/")                gives   "ABC"
HEAD(S, ALPHA)                  gives   "ABC"
TAIL(S, NOT "-")                gives   "-2+3"
HEAD(S, "123")                  gives   ""    % NULL STRING
TAIL(DROP(S, 7), "+-")          gives   "3"
```

THE REPEAT FUNCTION

<REPEAT function>

```
-- REPEAT -- ( -- <string expression> -- , -- <arithmetic expression> ->

>- ) ------------------------------------------------------------------|
```

If the value of <arithmetic expression> is zero, the result of the REPEAT function is the null string. If the value of <arithmetic expression> is greater than or equal to 1, the result is <arithmetic expression> occurrences of the <string expression> value concatenated together. If the value of <arithmetic expression> is less than zero, the result is an error.

Examples

```
REPEAT("ABC", 3)                gives   "ABCABCABC"
REPEAT(S, -5)                   gives   compile-time error
```

THE TRANSLATE FUNCTION

<TRANSLATE function>

```
-- TRANSLATE -- ( -- <string expression> -- , -- <translate table> ----->

>- ) ------------------------------------------------------------------|
```

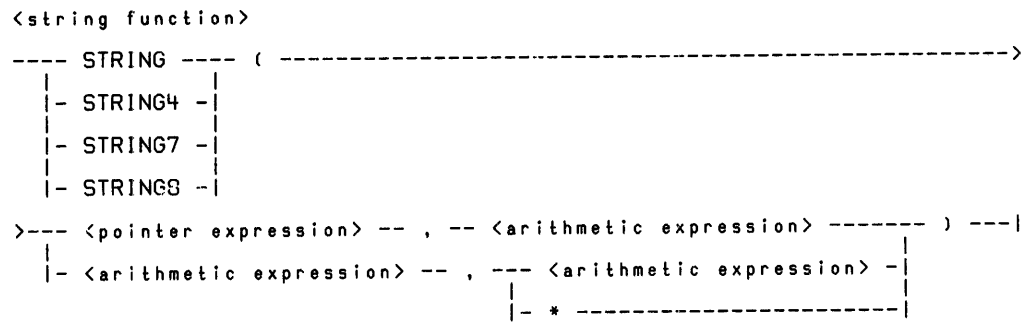
The result of the TRANSLATE function is a string of the same length as the <string expression>. Each element of the result string gets translated according to the <translate table>.

An explanation of the <translate table> can be found in the ALGOL Manual. A <translate table> may be a <translate table id> declared in the program or one of the standard supplied translation tables. The use of a <subscripted variable> as a <translate table> is not allowed.

Examples

```
TRANSLATE(S, HEXTOEBCDIC)
TRANSLATE(TAKE(S, 10), MYTT)
```

THE STRING FUNCTION



The STRING4, STRING7 and STRING8 functions return a hexadecimal string, an ASCII string and an EBCDIC string, respectively. The STRING function returns a string of the default character type.

The proper <string function> must be used to avoid illegal transfers.

The <string function> on a <pointer expression> provides for conversion from arrays to strings. The <string function> returns a string which is a copy of the characters pointed at by the <pointer expression>. The second parameter, rounded, specifies the number of characters taken.

The <string function> with <arithmetic expression> as the first parameter generates a new string whose value is the character representation of the first <arithmetic expression>. A formatting intrinsic is called to convert an integer, real or double number to an EBCDIC or ASCII string.

The number is converted into the most efficient form dependent on the length specified by the second <arithmetic expression> parameter as described in the following paragraph.

If the second parameter results in a value less than zero, an error will occur. If the value is equal to zero, the returned string is of length zero. If the value of this <arithmetic expression> is greater than the minimum number of characters needed to represent the first argument, a sufficient number of leading blanks are inserted. If the value is smaller than the number of characters needed, asterisks are put into the string. If the second parameter of the <string function> is an asterisk (*), the string will be just long enough to contain all the digits of the character representation of the first argument with no leading blanks. If the value of the first <arithmetic expression> is zero, the string will be one character long.

In the case of the STRING4 function, only the integer part of the first parameter will be converted. If the value of the second parameter is smaller than the number of digits in the resulting string, only the last digits will be returned. If the value of the second parameter is greater than the number of characters needed to represent the first argument, leading zero characters will be inserted.

Examples

```
STRING4(P, 20)
STRING7(POINTER(A), N - 3)
STRING(256, *)                  gives    "256"
STRING(-335.25, 8)              gives    " -335.25"
STRING(4.78●-2, 5)              gives    ".0478"
STRING(555000, 3)               gives    "6E5"
STRING(555000, 1)               gives    "*"
STRING4(456.789, 7)             gives    "0000456"
STRING7(185, -2.5)              compile-time error
```

THE CONCATENATION OPERATION
--- -------------- ---------

Two or more strings may be concatenated together by use of the CAT or || (double bar) operator. The concatenation of two strings yields a new string whose length is the sum of the lengths of the two original strings. The value of the new string is formed by joining a copy of the second string immediately onto the end of a copy of the first string.

Only strings of the same character type may be concatenated. If they are not of the same type, a syntax error occurs.

If more than one <string primary> is used in a concatenation operation, these are evaluated from left-to-right.

Examples

```
S1 := S2 || S3;
S1 := "AC" || S2 || "123";
S1 := S1 || TRANSLATE(S2, HEXTOEBCDIC);
```

THE BOOLEAN OPERATIONS ON STRINGS

The following BOOLEAN operators are available on operations with strings:

```
EQL   or   "="
NEQ   or   "¬="
```

Two strings are equal if and only if the LENGTHS are equal and every element in one string is equal to the corresponding element in the other string. Two null strings are equal.

```
GTR   or   ">"
GEQ   or   ">="
LSS   or   "<"
LEQ   or   "<="
```

One string is strictly greater than another string if and only if

1) The first differing element in a pair comparison of the two strings is greater in the first string than in the second string

   or

2) The LENGTH of the first string is greater than the LENGTH of the second and the two strings differ in no element up to the LENGTH of the second.

Only strings of the same character type may be compared; otherwise, a syntax error occurs.

Whenever two string constants can be compared as two arithmetic primaries, they are compared in that manner; i.e., the Boolean expression "AA" < "B" evaluates to FALSE because the two constants are considered to be arithmetic primaries (no change from pre-III.0 ALGOL). However, if two strings were assigned the values "AA" and "B" and the comparison made, the answer would be true; i.e., S:="AA"; T:="B"; S < T evaluates to TRUE.

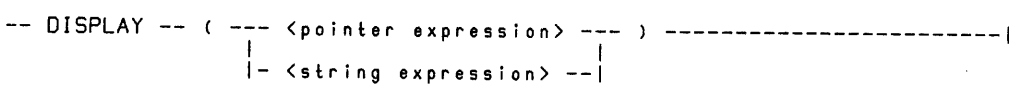
Examples

```
S1 := "AAB12+";
```

All these comparisons are true:

```
S1 EQL TAKE(S1, 6)
S1 NEQ HEAD(S1, ALPHA)
S1 GTR HEAD(S1, ALPHA)
S1 LSS DROP(S1, 3)
S1 LSS DROP(S1, 1)
```

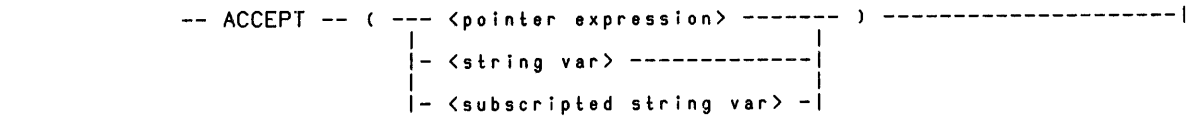
THE DISPLAY STATEMENT



A string may be used for displaying on the system SPO. The result of the <string expression> must be an EBCDIC string, otherwise, a syntax error occurs.

THE ACCEPT STATEMENT

ACCEPT Statement

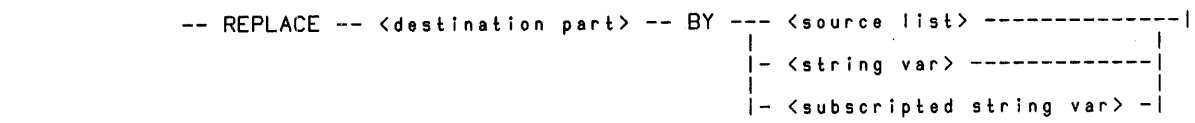


The <string var> or <subscripted string var> in an ACCEPT function must be of EBCDIC type; otherwise, a syntax error occurs.

The ACCEPT function displays its string parameter on the system SPO and waits for an operator response (via the AX message) as specified in the ALGOL Manual.

THE REPLACE STATEMENT

REPLACE Statement



To make it possible to transfer data from a string to an array, a <string var> or <subscripted string var> may be used as a <source list> in a REPLACE statement.

Examples

```
REPLACE P BY S;
REPLACE P BY SA[4, J];
```

THE LENGTH FUNCTION

<LENGTH function>

The LENGTH function is an integer function. The result is the number of characters in the string. The null string has the length zero.

Examples

```
S1 := "ABCDE";
LENGTH(S1) = 5;
LENGTH(EMPTY) = 0;
```

THE DECIMAL FUNCTION

<DECIMAL function>

This function returns as a double precision number the decimal value represented by the <string expression>. The double value is rounded to a single precision value when stored into a single precision (i.e., REAL) variable.

If the string does not represent a valid number, a run-time error occurs.

Examples

```
I := DECIMAL(S);
R := DECIMAL("-46.78E-3");
R := DECIMAL("-E5");
R := DECIMAL("A7");                    run-time error
```

EXTENSION TO PROCEDURE DECLARATIONS
--------- -- --------- ------------

A <string var> may be used as a formal parameter to a procedure. In this case, the <string var> must be specified in the procedure heading as a part of <formal parameter list>.

A <string var> may be used as a call-by-reference or call-by-value parameter. Any <string expression> may be passed as either a call-by-reference or call-by-value parameter. The default mode of passing a string is call-by-reference.

Note: The default mode of passing parameters in ALGOL is call-by-name, except when passing strings. Call-by-reference parameter passing has not previously been used as a method of passing parameters. The major difference between call-by-reference and call-by-name parameter passing has to do with the passing of expressions; i.e., passing something other than a simple variable or subscripted variable. When an expression is passed call-by-name, a "thunk" occurs; i.e., the expression is evaluated every time the parameter is referenced in the called procedure. In other words, a reference to the code to evaluate the expression is passed rather than the value itself. However, when an expression is passed call-by-reference, the expression is evaluated once at the time the expression is passed and the value of the expression is passed to the called procedure. Passing simple variables or subscripted variables call-by-name or call-by-reference works the same in both cases.

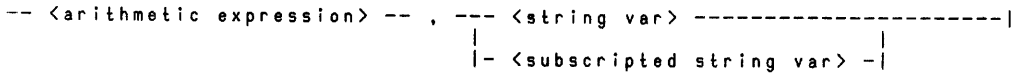
Examples

```
% PROCEDURE DECLARATIONS

    EBCDIC STRING PROCEDURE PROC(I,S);
    VALUE I; INTEGER I; STRING S;


    PROCEDURE P(S);
    VALUE S; STRING S;

% PROCEDURE CALLS

    P("ABCD"); P(TAKE(S, 10));
    P(STR || "10");
```

INPUT/OUTPUT

Unformatted I/O

<format and list part>



An <arithmetic expression> followed by a <string var> specifies that input data is to be assigned to the string variable without being edited. On output, the characters of the string variable are written as output without being edited. The number of characters read is determined by the maximum record size or the absolute value of the <arithmetic expression>, whichever is smaller. On output the number of characters is determined by the minimum of the maximum record size, the absolute value of the <arithmetic expression> and the string length.

Note that all counts are in characters, only if the file attributes UNITS=VALUE(CHARACTERS) and INTMODE=VALUE(SINGLE).

Examples

(Assume files "IN" and "OUT" have MAXRECSIZE >= 50 and UNITS=CHARACTERS).

```
READ(IN, 50, S);       % reads  50 characters into string S
WRITE(OUT, 25, S);     % writes 25 characters from string S
```

Formatted I/O

<string var> or <subscripted string var> are legal <list element>s for A, C and freefield format.

Strings behave in the same manner as "<pointer> FOR <arithmetic expression>" under the format phrases A and C.

Examples

```
READ(IN, <2A10>, S1, S2);
READ(IN, <A*>, 20, S);
WRITE(OUT, /, S1, S2, S[J]);
```

D2442 ALGOL - POINTER IN TRUTHSET FOR LENGTH

The syntax for the <table membership> form of <boolean primary> has been expanded to include:

<pointer expression> IN <table pointer> FOR <arithmetic expression>.

This allows the programmer to test whether the first <arithmetic expression> characters of the <pointer expression> are in the table referenced by the <table pointer>.

Example:

The following two statements have the same function as the third statement:

```
SCAN PTR FOR 10 WHILE IN ALPHA;
IF TOGGLE THEN . . .;

IF PTR IN ALPHA FOR 10 THEN . . .;
```

D2483 ALGOL - UPLEVEL POINTERS

Uplevel points and uplevel SWAP statements are now syntax errors. ALGOL no longer allows the POINTER FOR clause to be used for formal pointer parameters or in global declarations; e.g., the following example will now get a syntax error:

```
PROCEDURE PROC(PTR);
   POINTER PTR FOR GLOBALPTR;
```

Several problems with uplevel pointer syntaxing have been corrected. INTEGER and REAL functions using pointers as parameters will no longer cause incorrect uplevel pointer warnings. If a pointer assignment in a READLOCK function is uplevel, ALGOL will give a syntax error. Pointers in global declarations can only be assigned to themselves unless level 3 has been set. Also, any pointers, including formal pointer parameters, can be assigned level 2 pointers.

Example:

```
   BEGIN
        POINTER P1;
        PROCEDURE PROC;
        BEGIN
(1)          POINTER P2;
(2)          P1:=P2;          % This will give syntax error
             .
             .
             .
        END;
        .
        .
        .
   END.
```

In order to compile statement (2) syntax-free, change statement (1) as follows:

```
POINTER P2 FOR P1;
```

D2484 ALGOL - "DIRECTORYCONTROL, STACKSWAPPER" REMOVED

DIRECTORYCONTROL and STACKSWAPPER (not STACKSWAP) are no longer recognized as functions by the ALGOL compiler. The MCP procedures to handle these functions were removed on previous releases.

D2520 ALGOL - WAYS TO REMOVE USE OF "TOGGLE," "OVERFLOW"

The following examples indicate how to remove use of the TOGGLE and OVERFLOW functions.

TOGGLE Removal

Example of use of the TOGGLE function:

```
REPLACE PTR1 BY PTR2 FOR 10 UNTIL = ".";
IF TOGGLE THEN . . .;
```

A <residual count> can be used in the <count part> of the <replace statement>. Testing for the <residual count> = 0 will give the same result as testing TOGGLE.

The following statements have the same result as the TOGGLE example above.

```
REPLACE PTR1 BY PTR2 FOR N:10 UNTIL = ".";
IF N=0 THEN . . .;
```

Use of the "POINTER IN <truthset> FOR <length>" construct may also be used in lieu of TOGGLE. See ALGOL note D2442 for details.

OVERFLOW Removal

Example of use of the OVERFLOW function:

```
IF OVERFLOW THEN;                    % RESET THE OVERFLOW FLIP-FLOP
REPLACE PTR BY N FOR 3 DIGITS;
IF OVERFLOW THEN . . .;
```

Use of the OVERFLOW function can be removed by checking that the <arithmetic expression>, N, will fit in the number of digits specified.

The following statements have the same result as the OVERFLOW example above.

```
REPLACE PTR BY N FOR 3 DIGITS;
IF ABS(N) >= 999.5 THEN . . .;
```

D2563 ALGOL - INVALID CHARACTERS IN IDENTIFIERS

On the III.1 release, the use of identifiers with characters whose EBCDIC character code is in the range C1-FF which are not the letters A-Z and not the digits 0-9 will cause a syntax error. For example, AB\C is now a legal identifier, but will cause a syntax error on the III.1 release.

D2584 ALGOL - "SORT" STATEMENT

The ALGOL Reference Manual (No. 5001639), page 5-101, should be changed to read as follows:

<record length>

The <record length> represents the length, in words or characters depending on whether the array is a regular array or a character array respectively, of the largest item that is presented to the <sort statement>. If the value of <arithmetic expression> is not a positive integer, the largest integer that is not greater than the absolute value of the expression is used; that is, a record length of 12 is used if an expression has a value of -12.995. If the value of the <arithmetic expression> is 0, the program terminates.

ALGOL

P1103 ALGOL - ERROR FOR TOO MANY SUBSCRIPTS

The ALGOL compiler will now give the correct syntax error, rather than an INVALID INDEX, when a five-dimensional array with too many subscripts is referenced.

P1481 ALGOL - "THRU" LOOPS IN "I/O" LISTS

A THRU loop in an I/O list in a READ or WRITE statement that could affect later statements which involved constants has been corrected. For example, if a WRITE statement containing a THRU loop in the I/O list were followed by the assignment of a constant into a simple variable, code for the constant would not be emitted. This no longer occurs.

P1605 ALGOL - POINTER CASE EXPRESSIONS

The problem of erroneous "GLOBAL POINTER TO LOCAL NOT ALLOW" error messages, which could possibly be emitted following a pointer case expression, has been corrected.

P1627 ALGOL - CRITICAL BLOCK EXIT FROM "PROCESS" STATEMENT

It was possible for a program containing a PROCESS statement to get a critical block exit during execution if the critical block contained no other items which would require a call on the MCP procedure BLOCKEXIT. This has been corrected.

P1638 ALGOL - SCANNING AFTER "XREFFILES"

Dollar card options following XREFFILES will now be properly recognized.

P9072 ALGOL - "$INCLUDE" INTERRUPTS "$MAKEHOST"

When a host was compiled with $MAKEHOST, and procedures were added with the $INCLUDE option, a subsequent SEPCOMP into procedures following the included procedures would fail. Now, the $MAKEHOST option is reset at the beginning of an include and restored at the end.

P9101 ALGOL - REAL VALUED SERIAL NUMBERS

File declarations with real valued SERIALNO attributes now compile and execute correctly. Five- and six-character values for SERIALNO no longer cause the compiler to terminate with an integer overflow.

P9102 ALGOL - COMPILE-TIME DISPLAY

A compile-time display statement; e.g., 'DISPLAY . . ., which displayed more than 24 characters would lose some characters. This has been corrected.

P9211 ALGOL - "$LOADINFO"

Information files are now loaded correctly when $LOADINFO is set. A SEG ARRAY error will no longer occur during compilation.

ALGOL TABLEGEN

D2418 ALGOLTABLE - "ALGOL" TABLE GENERATOR

ALGOLTABLEGEN is used to create certain defines, arrays and value arrays for the ALGOL compiler. Specifically, ALGOLTABLEGEN creates the following:

1. The error message value array and error message defines
2. The defines for the ALGOL ELCLASSES
3. The defines for certain special values
4. The ALGOL INFO table entries for the identifiers recognized by the ALGOL compiler.

This program replaces TABLEGEN for the ALGOL compiler only. TABLEGEN is still available for the ESPOL and XALGOL compilers (see II.9 ALGOL note D2031 for a description of TABLEGEN). This description of ALGOLTABLEGEN is divided into two parts. The first part describes how to use ALGOLTABLEGEN; the second part describes the input data to ALGOLTABLEGEN.

How to Use ALGOLTABLEGEN

SYMBOL/ALGOLTABLEGEN is divided into two parts. The first part (sequence numbers 10000000-39999000) consists of all the current input data to ALGOLTABLEGEN. The second part (sequence numbers 40000000-end) consists of the actual program. The program requires the two following input files:

File CARD: This file contains all patches to the current input data.

File SYMBOL: This file contains the current input data which is to be patched. The default file name for this file is "SYMBOL/ALGOLTABLEGEN".

The two following output files are produced:

File PATCH: This file contains the patch to the ALGOL compiler created by running ALGOLTABLEGEN. The default file name for this file is "PATCH/ALGOL".

File LINE: This line printer file is created only if errors occurred during the execution of ALGOLTABLEGEN.

The following WFL deck will run ALGOLTABLEGEN:

```
<I> RUN SYSTEM/ALGOLTABLEGEN;
    DATA
    <patch cards>            % May include "%" comment cards and $ cards
<I> END.
```

If ALGOLTABLEGEN runs without errors, the following occur:

A patch will be created under most circumstances
The message "PATCH CREATED" will be displayed
MYSELF.TASKVALUE will be set to 1

If ALGOLTABLEGEN runs with errors, the following occur:

A patch file will not be created
The message "ERRORS-NO PATCH CREATED" will be displayed
MYSELF.TASKVALUE will be set to 0

If an updated SYMBOL/ALGOLTABLEGEN is required, a new symbol may be created by compiling the current symbol plus all patches and setting the NEW option. The following WFL deck will compile ALGOLTABLEGEN and create an updated symbolic:

```
<I> COMPILE SYSTEM/ALGOLTABLEGEN WITH ALGOL LIBRARY;
    ALGOL FILE TAPE=SYMBOL/ALGOLTABLEGEN;
    ALGOL FILE NEWTAPE=SYMBOL/NEW/ALGOLTABLEGEN;
    DATA
    $MERGE NEW
    <patch cards>
<I> END.
```

Note: The identical DATA deck may be used for both the execution and compilation of ALGOLTABLEGEN.

Input Data

As previously stated, sequence numbers 10000000-39999000 of SYMBOL/ALGOLTABLEGEN contain the current input data, which is divided into five parts. The following describes each of the five parts:

OPTIONS (10002000)

This section contains the name and beginning sequence number of each of the other four sections. This section should not be changed.

ERRORS (11000000)

This section contains the error message name (not to exceed 15 characters in length) and the error message text (not to exceed the remainder of the card image). The program will create a value array containing the text of all the error messages and will create defines for each error message. Each define contains the index into the value array and the character length of the error message text.

ELCLASSES (15000000)

This section contains a list of all the ALGOL ELCLASSES. Their numeric values are assigned in ascending order. These numeric values are the define values created for the compiler.

MISCELLANY (20000000)

This section contains a list of miscellaneous identifiers and the value which must be assigned to each identifier. As with the ELCLASSES, defines are created for the compiler using these identifiers and their specified values.

INFO (25000000)

This section contains a list of the identifiers recognized by the ALGOL compiler. The format of each record in the INFO section is described in detail in the symbolic at sequence numbers 25000000-25001100.

ATTABLEGEN

D2478 ATTABLEGEN - "WFL" JOBS SEE MORE TASK ATTRIBUTES

All WFL jobs can now test the value of the read-only task attributes ACCUMIOTIME and ELAPSEDTIME.

BACKUP

D2217 BACKUP - "AX-<RECORD #>" GTR CURRENT RECORD COUNT

BACKUP will now start printing from the start of a printer backup tape whenever a "<mix#> AX-<record #>" is entered that positions the tape past BOT.

D2275 BACKUP - DISALLOW "ND" WHEN USED WITH "*"

The use of "*" to cause BACKUP to print the backup files for a job before the job terminates is restricted to direct output only; the "ND" option may not be used with it.

Example:

```
CORRECT    - ?PB D * COPIES=4
INCORRECT  - ?PB D * ND COPIES=4
```

D2421 BACKUP - "RANGE" OPTION SEMANTICS

BACKUP has been modified to perform range checking differently. The following example will illustrate the change.

Suppose a printer backup file contains the following records:

Rec #	Contents
1	AAAA
2	BBBB
3	CCCC
4	AAAA
5	HHHH
6	DDDD
7	ZZZZ
8	DDDD

Suppose the following PB statements are used to print the above file:

```
1) PB <filename> KEY 1 4 RANGE "AAAA" "ZZZZ"
2) PB <filename> KEY 1 4 RANGE "AAAA" "DDDD"
3) PB <filename> KEY 1 4 RANGE "CCCC" "FFFF"
```

Previously, BACKUP ignored records within the range which had keys greater than the stop key. The PB statements would have printed:

```
1) Lines 1,2,3,4,5,6,7
2) Lines 1,2,3,4,6
3) Lines 3,4,6,8
```

BACKUP has been modified to treat records with keys greater than the stop key as if they had the stop key. Now, the PB statements will print:

```
1) Lines 1,2,3,4,5,6,7
2) Lines 1,2,3,4,5
3) Lines 3,4,5
```

D2443 BACKUP - DISK AND PACK SEARCHED FOR EACH FILE

BACKUP now searches disk and pack for each file it is asked to print. If a requested file is present on both families, it will be printed twice. If only the files under one of these families is to be printed, the following syntax must be used:

"<file name> ON <familyname>".

BACKUP has been further modified to search the system family "DISK" instead of the user's family.

D2583 BACKUP - "BACKUPBYJOBNR"

The discussion of BACKKUPBYJOBNR SYSTEM OPTION in "System Software Operational Guide", Volume 1, page 1-5-2 has been revised. The second paragraph has been corrected and now reads as follows:

> When this option is set, jobs are printed by order of the job number. When reset, jobs are printed in reverse order of print quantity.

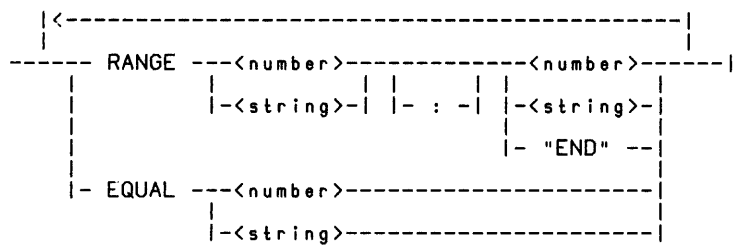
D2585 BACKUP - "END" AS RANGE STOP INDICATOR

The System Software Operational Guide, Volume 1, No. 5001563, should be corrected as follows:

Page 1-7-3

The syntax diagram for <range part> should be corrected to read as follows:

<range part>



Page 1-7-7

The semantic description of END should be corrected to read as follows:

"END" Is a range stop indicator for an EBCDIC string (besides either <number> or <string>) which is equivalent to setting the stop integer to 99999999.

BACKUP

P1001 BACKUP - EXECUTE "BACKUP" WITH "RUN" STATEMENT

BACKUP now works properly when executed directly with a RUN statement. The following two examples are equivalent:

```
PB D 157 SAVE COPIES 3
RUN SYSTEM/BACKUP("D 157 SAVE COPIES 3");
```

P1557 BACKUP - "QT" MALFUNCTION

BACKUP will now print the remaining BD files after one has been QTed.

P1565 BACKUP - PRINTER SELECTABLE WITH REEL # > "1"

PBing a backup tape, whose reel number is greater than 1, will now work correctly. Previously, the UNITNO attribute would be lost and the print job would go to any available printer.

P1568 BACKUP - ERROR FOR COLON WITHOUT "ALGOL" KEY

BACKUP would get a SEG ARRAY error for <keystart>=1 or 2 and would get no output for <keystart> >=3 when a colon separated start and stop values of RANGE. A colon should only be used if the key is ALGOL. BACKUP now gives an error and the message "COLON WITHOUT ALGOL KEY" while scanning the input specification.

P1639 BACKUP - UNQUOTED STRING AS RANGE VALUE

If a string used as a range value is not enclosed in quotes, an error will now occur.

P1640 BACKUP - PB "<FILENAME>" CP <UNITNO>

An erroneous SEG ARRAY error would occur when both a filename and a card punch unit were specified (e.g., PB "PBTEST" CP12). Now, this valid statement executes properly.

P9030 BACKUP - "BACKUP" VS. "GETSTATUS"

BACKUP will now handle backup files that are created under a usercode when the BDBASE and BDNAME options are used.

P9175 BACKUP - REMOVE EXTRA PAGE SKIPS

Extra page skips before and after BACKUP prints a file have been removed.

P9176 BACKUP - "SWAPPER" QT'S "BACKUP"

BACKUP will no longer be QT'ed by SWAPPER when running in swapspace.

P9198 BACKUP - NEW SYMBOLIC FOR "III.0"

The III.0 symbolic for BACKUP has been reformatted.

P9199 BACKUP - CLEAN UP ERROR HANDLING

BACKUP now terminates immediately upon encountering any syntax error in its input string, without attempting to print any backup files.

BASIC

D2280 BASIC - SCANIN REPLACED BY "TIMEINTRINSIC"

The BCL function will now invoke the MCP procedure TIMEINTRINSIC rather than executing the SCNI (scanin) operator. This change is made in the interest of hardware independence.

Due to the additional overhead in calling an MCP procedure rather than just executing an operator, the BCL function will take longer to execute.

D2456 BASIC - QUOTES IN "REMARK" MUST MATCH

Quotes appearing in a REMARK statement or in an apostrophe comment at the end of a statement must be matched.

BASIC

P1657 BASIC - SYNTAX ERROR FOR UNDECLARED IMAGES

Under certain circumstances, the compilers failed to give a syntax error for an image which was referenced but not declared in a program. If the program proved to be otherwise error-free, a codefile was produced which caused a system failure if executed. All undeclared images will now be flagged with error messages.

Another problem, which involved an INVALID INDEX in the compiler when only undeclared images appeared in a proram, has also been corrected.

P9104 BASIC - ERRONEOUS LINES WRITTEN TO "ERRORFILE"

The BASIC compiler will now write the erroneous line images to the ERRORFILE in addition to error messages. Previously, the compiler wrote only error messages which indicated the line number in error. Occasionally, the compiler displayed the wrong line number. In an effort to correct this failure and improve error diagnostics, the offending line will be written, followed by the appropriate message. The line number will appear with the text for that line, rather than appearing in the error message itself.

P9200 BASIC - SYNTAX "IF END/MORE" CORRECTLY

The IF END and IF MORE statements require a clause specifying a statement number for possible transfer of control. The correct syntax is:

IF <END/MORE> <file designator> THEN <statement number>

The compiler will now provide reasonable error messages when improper syntax is detected. Previously, syntax errors for these statements were misleading.

P9201 BASIC - EQUAL PRECEDENCE FOR DIVISION

Division now has the same precedence as multiplication, DIV, and MOD. Previously division had higher precedence, causing erroneous results in some expressions.

BINDER

D2424 BINDER - "AUTOBIND" AND "PPB"

If a code file were produced by both compiling through WFL with $AUTOBIND set and compiling or binding through CANDE with label equation, the code file would be bad. This problem has been corrected.

BINDER

P1466 BINDER - "$STRICT" GIVES SYNTAX ERRORS

Setting the STRICT dollar option now causes syntax errors for missing procedures specified on BIND cards. Previously, the STRICT option would only cause syntax errors if $LIST or $WARN were also set.

P9105 BINDER - LARGE HOST WITH SUBROUTINES

When binding FORTRAN subroutines to a host consisting of a main program and several subroutines, the BINDER would get an INVALID INDEX due to using the wrong figure from the BINDINFO to index the stack building code array. The BINDER now uses the correct figure.

P9106 BINDER - "USE" CARDS OBEYED

The BINDER would sometimes ignore USE cards following the binding of a procedure for which the BINDER itself had created a USE statement; e.g., "STATEMENT REQUIRED: USE XXXX FOR YYYY". The BINDER now faithfully obeys all USE cards.

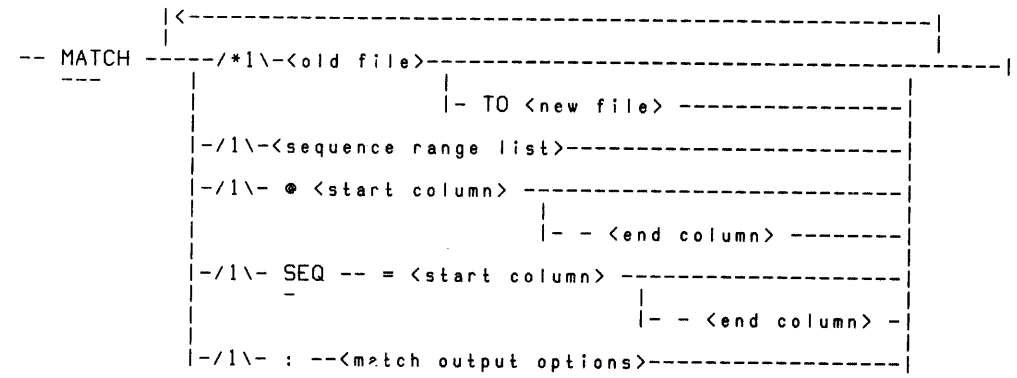
CANDE

D2281 CANDE - MANDATORY RECOVERYFILE FILEKINDS

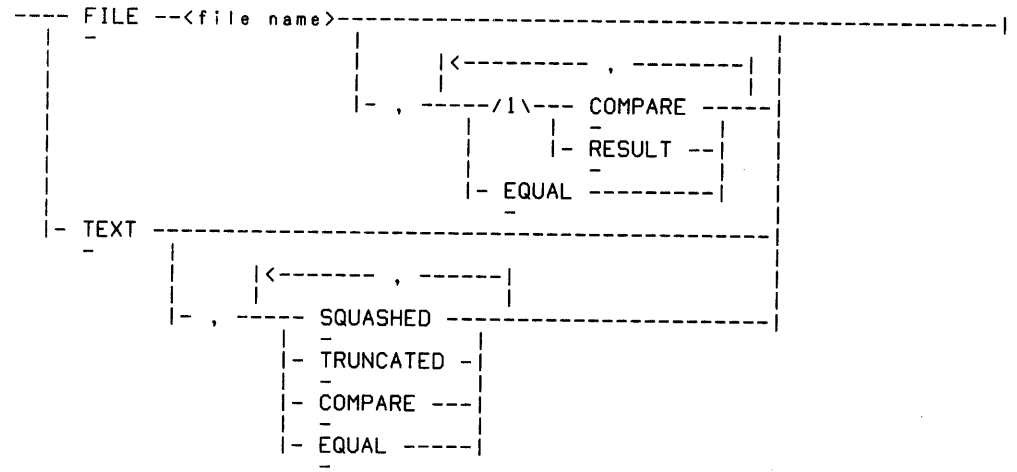
CANDE will no longer use a tankfile or recover a recovery file that was created by a pre-II.8.1 CANDE. Also, these files must now have a filekind of RECOVERYFILE to be considered valid.

D2444 CANDE - "MATCH" VERB

The MATCH command compares two files and determines the differences between them. Output indicating the result of the comparison may be directed to the terminal or to a new file. The syntax is as follows:



<match output options>



SEMANTICS:

The MATCH command causes the <old file>, or portions thereof, to be compared with <new file>. If a <new file> is not specified, the workfile is used by default.

For sequenced files, if a record is encountered in each file having the same sequence number, a character by character comparison of the records is done. If the two records are found to differ, the <new file> record will be output as a replacement line. If a record exists at a sequence number in <old file> but does not exist in <new file>, that record will be output as a deleted line. If a record exists at a sequence number in <new file> but not in <old file>, that record will be put out as an inserted line.

For unsequenced files, the first record of <old file> will be compared character for character with the first record of <new file>, the second with the second, et cetera.

Any two files that are MATCH'ed must be of the same type.

If a <sequence range list> is specified, the files will only be compared within that sequence range. If a column specification is used (e.g., * <start column>, etc.), the records will be compared within that column range. If the "SEQ" option is used, the indicated column range will be used as sequence numbers for the comparison. The width of this sequence number may not exceed eight digits, and the option may only be used with DATA files.

The results of the comparison are formed according to the output options used. By default, output is directed to the terminal and lists only the sequence numbers. Line deletions are preceded by "-", line replacements by "R", and line insertions by "I".

Two output options are available:

1. TEXT

   Output is directed to the terminal, with line deletions preceded by "-", replacements by "R", and insertions by "I".

   If the COMPARE option is specified, the results are similar to that of the "COMPARE" option in the LIST command.

   If the EQUAL option is used, records which are the same will be listed.

   If the SQUASHED option is used, any group of multiple blanks is reduced to a single blank for output of the record.

   If the TRUNCATED option is used, a line too long for the terminal is truncated to one terminal line.

2. FILE <filename>

   Output is directed to the specified file. The contents of the file are the same as those given for TEXT. The resulting file will be of type DATA.

   If the COMPARE option is used, the contents of the file are the same as those given for the use of COMPARE with the TEXT option.

   If the RESULT option is used, the result file will be the same type as that of the files MATCH'ed.

   For language source files, line insertions and replacements are put into the result file. Single line deletions are replaced by a "$" record. Multiple line deletions in sequence will cause output of a "$SET VOIDT" record at the starting sequence number and a "$POP VOIDT" at the ending sequence number.

   For DATA file comparisons using the sequence number column specification, line replacements and insertions are put into the result file. For deletions, a blank record is put into the file with sequence number intact.

   For DATA file comparisons not using the sequence number column specification, when records differ, the <new file> record is placed in the result file. If <new file> is longer than <old file>, trailing unmatched records are placed in the output file. If <old file> is longer than <new file>, the trailing unmatched records are not placed in the output file.

   If the EQUAL option is used, records which are the same will be placed in the file.

EXAMPLES:

MATCH ALPHA

MATCH the file ALPHA to the workfile, listing only sequence numbers.

MATCH WILDCATS TO OWLS ON TV29

MATCH the file WILDCATS to the file OWLS on the pack TV29, listing only sequence numbers.

MATCH #20-30 A/B

MATCH the file A/B to the workfile, matching only columns 20 through 30 inclusive, listing only sequence numbers.

MATCH A TO B SEQ=10-17

MATCH the DATA file A to the DATA file B; the sequence numbers will be found in columns 10 through 17 inclusive.

MATCH YOURS TO MINE :F OURS

MATCH the file YOURS to the file MINE, creating a DATA file called OURS which will contain the differences between the two files.

MATCH OLDSOURCE TO NEWSOURCE : F FIX, RESULT

MATCH the file OLDSOURCE to the file NEWSOURCE, creating a file called FIX which will contain source records for replacements and "VOIDT" or "$" cards for deletions.

MATCH :T BAD TO GOOD

MATCH the file BAD to the file GOOD, listing differences.

MATCH FLYERS TO ISLANDERS:T,C

MATCH the file FLYERS to the file ISLANDERS, listing differences using the COMPARE form.

MATCH @20-30 Z1 TO Z2 :T,T,SQUASHED,COMPARE

MATCH the file Z1 to the file Z2, matching only columns 20 through 30 inclusive, listing differences, truncating where necessary and replacing all multiple blanks by a single blank.

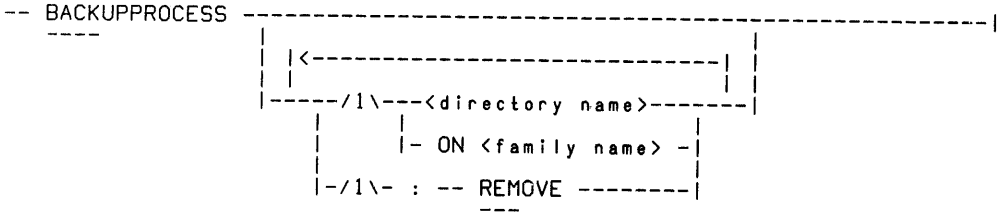
D2508 CANDE - "CANDE" BACKUP FILE PROCESSING FACILITY

The BACKUPPROCESS command will allow a user to create disk file copies of, identify, remove and list backup files. Backup files a user may access include backup files created in his session on DISK and PACK under the *BD and *BP directories, backup files created in his session on any arbitrary disk family, and backup files in his directory. BACKUPPROCESS contains all of the features of the BDREMOVE command. BDREMOVE will be de-implemented on a future release.

As in the BDREMOVE command, each backup file referenced is presented to the user. Proper responses to a presented backup file are NEXT, REMOVE, WHAT, COPY and LIST.

The CANDE BACKUPPROCESS command is implemented as a separate task, CANDE BACKUPPROCESSOR, which communicates with the user via a remote file. Termination occurs when all backup files have been presented, an end-of-file indication for the remote file is detected or the BACKUPPROCESS task is DS'ed.

BACKUPPROCESS Command Syntax



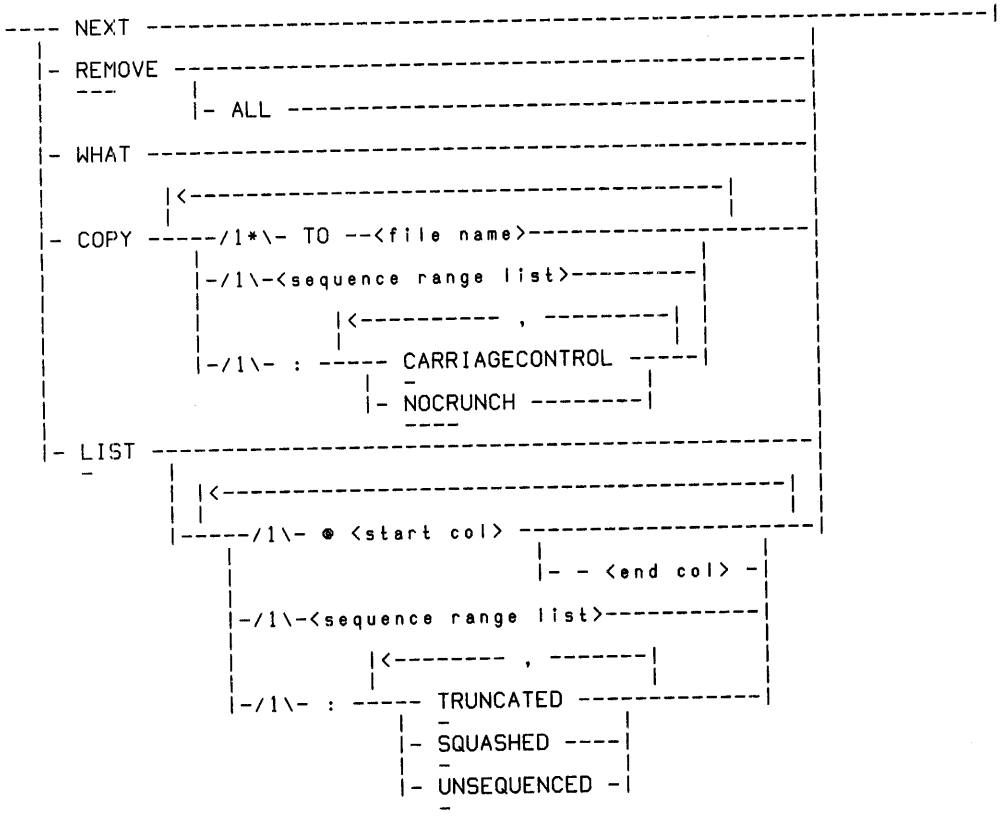
BACKUPPROCESS Command Semantics

If REMOVE is specified, all backup files referenced will be removed without being presented.

<directory name> specifies a directory in the user's library that is to be searched for backup files. Only backup files found in this directory are presented to the user. The directory may only designate files in the user's library.

ON <family name> allows the user to specify an arbitrary disk family to be searched for *BD and *BP files created in his session.

CANDE BACKUPPROCESSOR Syntax



CANDE BACKUPPROCESSOR Semantics

NEXT

NEXT will cause the next file to be presented.

REMOVE

REMOVE will cause the presented file to be removed. If ALL is specified, the remaining files will be removed without being presented; otherwise, the next file is presented.

COPY

COPY is used to create a disk file copy of the presented backup file. This disk file is created in the users's directory on his primary disk family with the name <file name>. An error will be issued and the COPY will not performed if <file name> is in use by an existing file.

The disk file created will have a MAXRECSIZE of 30 words, a BLOCKSIZE of 270 words and an AREASIZE of 504. Each record from the backup file will be translated to EBCDIC, if necessary, and padded on the right by blanks to fit a 30 word record.

An entry in the backup file with "SPACE 2" carriage control will be treated as two records. If the entry contains no text, two blank records will be inserted in the disk file. If the entry contains text, one blank record will be inserted in the disk file. All other entries will be counted as one record. If they do not contain text, a blank record will be inserted into the disk file.

The CARRIAGECONTROL option causes carriage control information for BD files to be encoded in each disk file record in CTL360 format. This means that the first character of each record will contain carriage control information with data beginning in the second. Since blank records are inserted as described above, only "SKIP" and "SPACE 0" carriage control information will be encoded. The filekind of the disk file will be made BDDATA, (value = 191), to allow easy external recognition. CANDE WRITER will use the encoded carriage control of BDDATA files and not reformat them, unless a column range is specified which does not contain column 1. With this option, the representation of the BD file, or portions thereof, may easily be recreated if desired. If the file is a BP file, the CARRIAGECONTROL option will be ignored.

If the CARRIAGECONTROL option is not used or the file is a BP file, the first character of each disk file record will be the first character of data and the filekind of the disk file will be made DATA.

The NOCRUNCH option will cause the disk file to not be crunched. If this option is not specified, the disk file will be crunched.

The meaning of <sequence range list> for a backup file is analogous to the meaning of <sequence range list> for a disk file whose filekind is DATA. This means that CANDE will assign a "sequence number" equal to the record number multiplied by 100 (i.e. 100,200,300....).

Note: Sequence ranges are not allowed with old backup files.

The same backup file will be re-presented after it is copied.

WHAT

WHAT will display the title of the presented backup file and the name of the job that created it. If the USERCODE and/or the CHARGECODE of the job that created the backup file are different from those of the current session, they will be displayed also. The number of records contained in the backup file will also be displayed if such information is available.

LIST

LIST will list the presented backup file, or portions thereof, at the user's terminal, translating each non-graphic character to an EBCDIC question mark, "?".

An entry in the backup file with "SPACE 2" carriage control will be counted as two records. If the entry contains no text, two blank records will be listed. If the entry contains text, one blank record will be listed. All other entries will be counted as one record and listed as such. Entries that do not contain text will be listed as blank records. Each record listed will be preceded by its "sequence number", (unless the :UNSEQUENCED option is used), as if a data file were being listed with the normal CANDE LIST command.

Note: Sequence ranges are not allowed with old backup files.

This scheme of record counting and sequence numbering means that a disk file which is created by the COPY option and listed later with a normal CANDE LIST command will appear exactly as the original backup file would appear when listed with the BACKUPPROCESS LIST option.

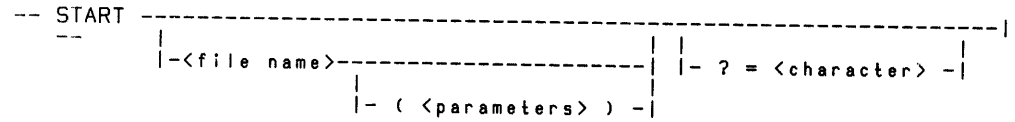
The column range specifications, and the TRUNCATED, SQUASHED and UNSEQUENCED options function exactly as for the normal CANDE LIST command. The <sequence range list> is as described for the COPY option.

The same backup file will be re-presented after it is listed.

D2509 CANDE - PARAMETERS TO "START" JOBS

The ability to pass parameters to CANDE START jobs has been implemented.

Syntax:



<parameters> is any arbitrary list of constants (i.e., integer, real, string or Boolean) separated by commas recognized by WFL. See WFL note D2514 for a discussion of job parameters.

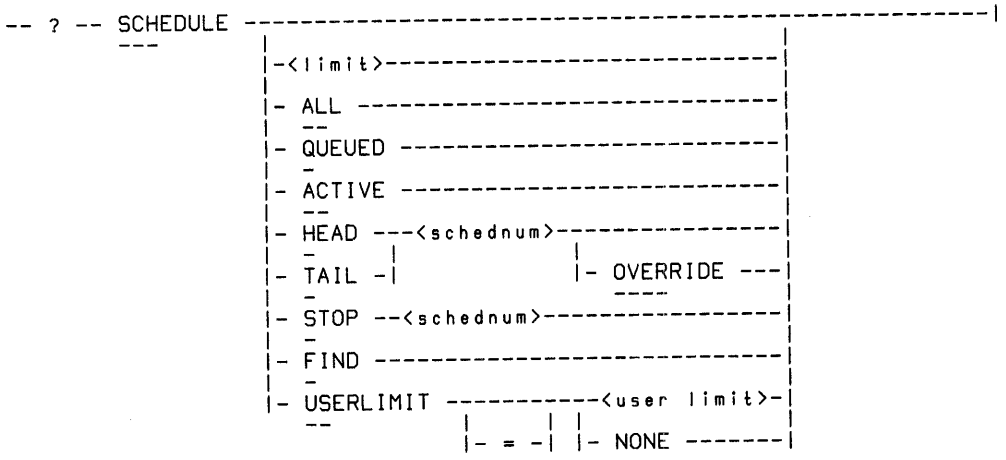
D2544 CANDE - LISTING TERMINATION WITH "BREAK ON OUTPUT"

CANDE will now terminate any listing function (i.e., LIST, FIND:T, FILES, etc.) upon receipt of a Class 99 message with the "break on output bit" on if there were no message to recall.

D2552 CANDE - USERLIMIT FOR SCHEDULE SESSIONS

A new option, USERLIMIT, has been added to the ?SCHEDULE operator control command. This option limits the number of CANDE SCHEDULE sessions any user may have active at one time, thus preventing one user from clogging the SCHEDULE mechanism with many long sessions to the exclusion of others who have only a few small sessions.

Syntax:



If <user limit> is specified, it must be a non-zero unsigned integer number. It may be larger than <limit>; however, if it is larger than the maximum number of SCHEDULE stations CANDE will service, CANDE will lower <user limit> accordingly. Specifying a USERLIMIT other than NONE will prevent CANDE from starting a SCHEDULE session for any user who already has <user limit> or more SCHEDULE sessions currently active. SCHEDULE sessions in excess of <user limit> will remain scheduled until a current session for that user finishes, or the operator raises the value of USERLIMIT.

When USERLIMIT is set to NONE, CANDE will start SCHEDULE sessions in the order that they are scheduled without regard to how many schedule sessions are already running for any user. This is the manner in which SCHEDULE sessions were started prior to III.0 CANDE.

The value of USERLIMIT is saved in CANDE's TANKFILE and preserved over a halt load. NONE is the default setting for USERLIMIT when creating a new TANKFILE or using a TANKFILE created by a pre III.0 CANDE.

Example:

```
?SCH
#SCHEDULE LIMIT=5 NO USERLIMIT
?SCH USERLIMIT = 1
#USERLIMIT=1
?SCH
#SCHEDULE LIMIT=5 USERLIMIT=1
      :
      :
      :
?SCH
#SCHEDULE LIMIT=5 USERLIMIT=1 ACTIVE=3 READY=3
?SCH ALL
#00054 SN=3884 USERA
#00060 SN=4217 USERB
#00059 SN=4197 USERC
#00055 WAITING USERA
#00056 WAITING USERA
#00057 WAITING USERA
#SCHEDULE LIMIT=5 USERLIMIT=1 ACTIVE=3 READY=3
?SCH USER 2
#USERLIMIT=2
?SCH ALL
#00054 SN=3884 USERA
#00060 SN=4217 USERB
#00059 SN=4197 USERC
#00055 SN=4311 USERA
#00056 WAITING USERA
#00057 WAITING USERA
#SCHEDULE LIMIT=5 USERLIMIT=2 ACTIVE=4 READY=2
```

CANDE

P1002 CANDE - STORE LENGTH OF "FIND/REPLACE"

CANDE was failing to enter the length of a FIND/REPLACE command in the scratch block, sometimes causing both a tank block to be destroyed and CANDE to loop. This has been corrected.

P1104 CANDE - EMPTY BUFFER SIZE IN "TERMINAL" COMMAND

CANDE will now correctly handle options following a BUFFER option with an empty size specification in the TERMINAL command.

Example 1

```
TERM B S
```

CANDE will now set "SCREEN" rather than ignore it.

Example 2

```
TERM B P 24
```

CANDE will now set the page size to 24 rather than give the error "SYNTAX ERROR. SCANNING 24".

P1105 CANDE - NONGRAPHIC CHARACTERS IN "?SS" AND "?TO"

CANDE will now translate all nongraphic characters (except BEL, 4"2F") in "?SS" and "?TO" commands to the EBCDIC "?" character.

P1106 CANDE - COMPILE TIME TASK ATTRIBUTES

CANDE will now give a syntax error for compile-time task attribute assignments which cannot be passed to a compiler. Previously, these assignments were ignored.

P1107 CANDE - INCORRECT STATISTICS

An inconsistency in CANDE's internal statistics accumulation mechanism has been corrected.

P1234 CANDE - CONFUSION WITH "?REPEAT" AND "?REPORT"

CANDE will no longer interpret a ?REPORT command in a SCHEDULE session as a ?REPEAT command.

P1235 CANDE - BEFORE OPTION IN "?ENTER" VERB

The BEFORE option of the ?ENTER verb now works correctly. The item being entered will no longer cause another item to be deleted.

P1497 CANDE - "?QUIT" FAILURE

A problem with ?QUIT failing to cause CANDE termination has been corrected.

P1498 CANDE - INCORRECT DESTINATIONS FOR "?REPORT" MESSAGES

If two usercodes were greater than 5 characters in length and differed ony in the last character, ?REPORT messages from either would be received by both. This has been corrected.

P1499 CANDE - INCORRECT ERROR MESSAGES

A problem with incorrect error messages being issued due to duplicate entries in an internal value array has been corrected.

P1500 CANDE - STATE DUMP PROBLEMS

A problem with recurring state dumps has been corrected.

P1501 CANDE - "REP, RET, TAKE, ENTER" WITH "CONTINUE"

If a CANDE command contained a continuation character when no character had been assigned, CANDE would respond with an error message. If the CONTINUE command were then used to specify a continuation character, which was followed by a RETRIEVE and REPEAT of the original command which was in error, CANDE would fail to remove the continuation character from the text of the command, causing another error. This problem has been corrected.

P1502 CANDE - LONG FILE EQUATIONS WITH COMPILES

CANDE will now resize the sheet array passed to compilers to handle arbitrarily long lists of file equations. This prevents such problems as a SEG ARRAY error in the ALGOL compiler.

P1503 CANDE - SPECIAL CHARACTERS IN "UTILITY" TEXT

Any special character may now be passed to a user utility program via the UTILITY command. Arbitrary restrictions against these characters, such as backslash ("\") have been eliminated.

P1656 CANDE - LOST RECORDS WITH "MOVE" OR "INSERT"

CANDE will no longer lose workfile records with a MOVE or INSERT command.

Examples:

```
MOVE <s1> - END TO <s2> + <inc>
INSERT <s1> - END AT <s2> + <inc>
```

The above examples will no longer lose the record from <s2> to the end of the file if <s1> is larger than the sequence number of the last record of the file.

P9107 CANDE - INCORRECT "GRINDLIMIT" SETTINGS

CANDE will no longer process more GRIND stacks than specified with the GRINDLIMIT command.

P9108 CANDE - "PRIVATE" SECURITY IN "MAKE" COMMAND

CANDE will now correctly set the security type of the workfile to PRIVATE if PRIVATE is specified in the MAKE command.

P9300 CANDE - ELIMINATE "HOLDQ" PROBLEMS

Problems with the HOLDQ mechanism involving asynchronous tasks which have done a foreign file open on the station have been eliminated.

COBOL

D2259 COBOL - DECIMAL ACCURACY IN COMPUTE STATEMENT

An improved method of analyzing arithmetic expressions in compute statements is now available as a default which produces optimal code for evaluating compute statements and improves the decimal accuracy of results from arithmetic expressions containing division operations. This method does not produce the greater time and memory requirements associated with the previous method which was enabled only if the compiler was compiled with the compile-time option "TREEDRIVEN" set. The previous method and the "TREEDRIVEN" option have been removed from the compiler.

D2284 COBOL - "SIGN" CLAUSE

The specification of "TRAILING SEPARATE CHARACTER" signs for usage DISPLAY items is now allowed.

D2500 COBOL - GLOBAL DATA BASES

The order in which stack locations were assigned for data bases was different between the II.9 ALGOL and COBOL compilers, preventing mixed language binding of programs accessing global data bases.

Consequently, a change has been made in the order in which the COBOL compiler assigns stack locations for data bases, requiring recompilation of all COBOL programs accessing global data bases which will be bound together.

D2521 COBOL - FLOATING POINT DATA DECLARATIONS

The usage declarators, COMP-4 and COMP-5, are now prohibited in the data description entry of a group item; they are allowed only for elementary items.

Previously, the presence of a COMP-4 or COMP-5 declaration at the group level did not cause the COMP-4/COMP-5 usage to "percolate down" to subordinate elementary items having no explicit usage specification as other usages do, but did cause these elementary items to be considered as usage computational. This action was extremely misleading and caused error messages when the PICTURE clause was omitted from the data description entry of the elementary items.

D2522 COBOL - STATISTICS DOLLAR OPTION

When the "STATISTICS" dollar option is set, the compiler generated code for timing purposes now uses the "TIME(12)" function (processor time) rather than the "TIME(11)" function (elapsed time).

COBOL

P1003 COBOL - PROGRAM PORTABILITY

Messages warning of program non-portability are now issued for the declaration of BCL internal data or for compiling a program at a lex level above 15.

P1004 COBOL - BLOCK NAME FOR "$STATISTICS" CODE LISTING

The listing for $STATISTICS now identifies the associated block (paragraph) name by including the name with the number. Previously, the listing said "ACTIVE CLOCK".

P1005 COBOL - RESERVED WORD IN ARITHMETIC STATEMENT

Some reserved words used incorrectly in an arithmetic statement; e.g., SUBTRACT INDEX FROM INDEX, no longer cause the compiler to discontinue with an INVALID INDEX.

P1006 COBOL - "INVALID INDEX" ON "WAIT" STATEMENT

The COBOL compiler no longer gets an INVALID INDEX while attempting to compile an erroneous WAIT statement.

P1007 COBOL - OPTIMIZED "PERFORM" STATEMENTS

A problem with in-line perform ranges, which occurred in rare instances when "OPTIMIZE" was set, has been corrected and no longer causes spurious syntax errors.

P1043 COBOL - NUMERIC TEST FAILED TO CHECK SIGN FIELD

COBOL numeric test now checks on the sign field for COMP-2 items.

P1044 COBOL - INCORRECT TIME GENERATED BY "ACCEPT"

The COBOL ACCEPT FROM TIME statement now generates the correct time in the hundredths of seconds places.

P1088 COBOL - SEMICOLON IN "NEXT SENTENCE" AND "LOCK"

In the following COBOL statement, the semicolon after the NEXT SENTENCE phrase was erroneously not allowed:

IF <condition>; THEN NEXT SENTENCE; ELSE STATEMENT-2

Also,
the semicolon preceding the AT in the LOCK statement was
erroneously not allowed. This has been corrected.

P1089 COBOL - "INVALID INDEX" BY "LOCK" STATEMENT

The LOCK statement no longer gives an INVALID INDEX in the compiler.

P1090 COBOL - "RERUN" STATEMENT BRANCHED INCORRECTLY

Branch in RERUN code now branches to the correct code syllable; previously, it would delete the stack word needed for an I/O test.

P1091 COBOL - CODE LISTING

The readability of the code listing has been improved.

P1092 COBOL - PREVENTION OF INTEGER OVERFLOW

A procedure has been added to the compiler to check a numeric literal before trying to use the INTEGER function to convert to an integer.

P1093 COBOL - XREF GIVING WRONG SEQUENCE NUMBERS

The Xref list had items listed as occurring on lines where they did not occur as well as listing the lines on which the items did actually occur. This has been corrected; there is now one listing for each actual occurrence.

P1145 COBOL - RECOGNITION OF NUMERIC LITERALS

The compiler is no longer confused when a non-integer numeric literal appears in the rightmost positions of a card image and the next card image is a non-continuation card whose first non-blank character is an "E".

P1146 COBOL - INVALID LEVEL NUMBERS

The specfication of illegal level numbers no longer can cause the compiler to abnormally terminate.

P1147 COBOL - "PERFORM" STATEMENTS

The correct name of the offending procedure-name is now listed with error messages relating to invalid transfer of control for PERFORM statements.

P1148 COBOL - "COPY" STATEMENTS

The compiler no longer fails to properly handle COPY statements specifying internal library names with an embedded hyphen.

P1149 COBOL - "XREF"

References to data items by REDEFINES and RENAMES clauses are no longer omitted from the cross reference report.

P1150 COBOL - PROGRAM COLLATING SEQUENCE CLAUSE

When a program collating sequence clause is present in the object-computer paragraph and the supplementary alphabet-name declaration is missing, the compiler no longer fails to emit a syntax error.

P1151 COBOL - UNNECESSARY CODE GENERATION

Unnecessary NOOP operators are no longer occasionally emitted prior to NAMC operators referencing character descriptors.

P1152 COBOL - UNDIGIT LITERALS IN "VALUE" CLAUSE

The compiler no longer terminates with an INVALID INDEX when attempting to compile a syntactically invalid VALUE clause that specifies an undigit literal in the data description of an item declared subordinate to an item with an OCCURS clause.

P1158 COBOL - GENERALIZED FILE SPECIFIER IN "USE" SECTIONS

The compiler no longer fails to detect an illegal perform of a part of a USE section in the declaratives that references the generalized file specifier.

P1159 COBOL - "TIME" INTRINSIC

The compiler will no longer fault with an integer overflow if a large number is used as an argument to the TIME intrinsic.

P1160 COBOL - LISTING OF OMITTED SOURCE LANGUAGE LINES

The compiler no longer fails to list lines that have been omitted.

P1237 COBOL - SIZE ERROR CODE OVERWRITTEN

The compiler no longer overwrites good code used to check size error in ADD. Initialization code for size error checking was being overwritten, allowing spurious size error condition code to be executed.

P1240 COBOL - "XREF"

References to data items in the USING clause are now listed in the cross reference report.

P1282 COBOL - "ADD" STATEMENTS

Incorrect code is no longer sometimes emitted when a data-name subscripted by a constant is used as a source operand in an ADD statement.

P1285 COBOL - LITERALS IN LOCAL STACK

Certain literals were stored in the D1 stack for efficiency; however, binding multiple separately-compiled programs wastes space with duplication of stored literals. Now, any programs compiled at level 3 or higher will store these literals in the local stack for that program.

P1380 COBOL - CONDITION NAMES

The code generated to evaluate condition-name conditions for variable length conditional variables has been corrected.

P1411 COBOL - "MONITOR"

Paragraph names listed in a MONITOR statement, but never occurring in the procedure division, no longer cause the compiler to abnormally terminate with an INVALID INDEX.

P1413 COBOL - "SET" STATEMENTS

Non-integer numeric literals are no longer acceptable as source operands in SET statements.

P1504 COBOL - "STATISTICS" OPTION

Timings for interrupt procedures are now included in the summary report produced by the STATISTICS option.

P1507 COBOL - SYNTAX GROUP COMP ITEM USED AS SORT KEY

An item declared as group computational should not have been allowed to be used as a key to the sort intrinsic. The compiler now syntaxes any attempt to use a group computational item for a sort key.

P1510 COBOL - GLOBAL DIRECT FILES PASSED TO "ISAM"

Epilogue procedures no longer cause global direct files to be closed.

P1538 COBOL - SET DYNAMIC FILE ATTRIBUTES OF "SORT" FILES

Files with dynamic attributes, when used by SORT, were not getting those attributes set at open and close time during a SORT. These attributes are now set when sorting.

P1539 COBOL - MOVE SCALED INTEGER TO ALPHANUMERIC DISPLAY

An item declared as an integer with assumed decimal scaling (edit character 'P') may now be moved to items declared alphanumeric display. Formerly, this was erroneously disallowed.

P1540 COBOL - INDEX DATA NAMES

Invalid offsets are no longer calculated for display data items declared subsequent to index data items which are, in turn, declared subsequent to odd-length COMP-2 data items in a display record.

P1567 COBOL - "ANSI74" DEFAULT WRITE AFTER ADVANCING

The compiler defaulted to "WRITE BEFORE ADVANCING" when the ADVANCING clause was not explicitly stated. The ANSI74 standard states the default should be "WRITE AFTER ADVANCING". Under the ANSI74 dollar option, the compiler will default to "WRITE AFTER ADVANCING"; without this dollar option set, the compiler will default to "WRITE BEFORE ADVANCING".

P1604 COBOL - LINAGE LINES CLAUSE LEFT "I/O" RESULT ON STACK

I/O done to satisfy ANSI74 linage lines clause was leaving I/O result on top of stack, which could cause spurious errors, especially in performs. This has been corrected.

P1607 COBOL - REPORT WRITER

Record areas obtained for report files are no longer sometimes larger than necessary.

P1641 COBOL - VALUE CLAUSE

The compiler no longer incorrectly allows the value clause in the data description entry of a data item in a record description for a sort file when the B2500 system option is set.

P1648 COBOL - ILLEGAL SYNTAX FOR "READ" NOT CAUSING ERROR

The COBOL READ statement does not allow the following syntax:

READ <filename> NEXT KEY IS <data-name>

Previously, the compiler failed to give a syntax error for this; now, the compiler syntaxes this as an unrecognized construct.

P1649 COBOL - "NEXT SENTENCE" IN "INVALID KEY"

The ANSI74 Standard specifies that the INVALID KEY phrase be followed by an imperative statement. It does not allow the NEXT SENTENCE phrase as is the Burroughs extension without the ANSI74 dollar option. The tree-driven code, activated by the OPTIMIZE dollar option or by compiling the compiler with the TREEDRIVEN dollar option, erroneously still allowed this extension. Now with the ANSI74 dollar option, the NEXT SENTENCE phrase is always syntaxed.

P1650 COBOL - "OPEN EXTEND" LEAVING "I/O" RESULT ON STACK

An OPEN EXTEND statement inside a perform range no longer causes a perform range to fail to return, since an I/O result was being left on the stack.

P1654 COBOL - SEQUENCE DOLLAR OPTION NOT READING INCREMENT

The SEQ dollar option with increment was being syntaxed incorrectly; e.g., it would disallow "$SEQ 100+100". This has been corrected.

P1655 COBOL - "ACCEPT" USING INCORRECT QUALIFICATION

ACCEPT referenced the receiving data item improperly, so that the correct qualification would not always occur. The ACCEPT referencing by the compiler has been changed to ensure proper qualification.

P9067 COBOL - CONSTANT SECTION COMPUTATIONAL DATA ITEMS

Incorrect code is no longer generated for all references to constant section computational items which had been used as sending operands in a MOVE statement to an alphanumeric data item.

P9111 COBOL - "STRING" GET ERRONEOUS SYNTAX ERROR

The COBOL compiler no longer gives an erroneous syntax error on the STRING statement when a semicolon is included before the ON OVERFLOW phrase.

P9112 COBOL - GROUP COMPUTATIONAL DATA ITEMS

Legitimate references to character-oriented group computational data items no longer cause an INVALID INDEX at run time because character offsets are used to index word descriptors.

P9194 COBOL - MULTIPLE "COPY" STATEMENTS

The library file kind attribute has now been reset to its default value of "VALUE(DISK)" so that a "COPY" statement not explicitly referencing a family does not request files on a family specified by a previous COPY statement.

P9195 COBOL - INCLUDE SIZE IN GLOBAL ARRAY ITEM DESCRIPTION

COBOL no longer fails to include the size of a global array in the item description in the program description.

P9299 COBOL - "SEG ARRAY" TERMINATION OF COMPILE

A potential cause of a SEG ARRAY terminaton has been eliminated when compiling a large library with numerous "88" levels.

CONTROLLER

D2267 CONTROLLER - FOUR VS THREE DIGIT SYSTEM SERIAL NUMBERS

All system serial numbers have been expanded to allow four-digit numbers.

CONTROLLER

P1336 CONTROLLER - "MCS" MESSAGE CAUSES "SEG ARRAY"

The CONTROLLER will now check the length of the USERCODE in an input message from an MCS. If the USERCODE is greater than 17 characters in length (illegal USERCODE syntax), the message will be discarded.

P1399 CONTROLLER - "DIR" VS. INPUT SCANNER

The CONTROLLER will now fire off SYSTEM/FILEDATA correctly when the following request is made:

"DIR NAME= <packname>".

If more than one "blank" appears on either side of the equals side, the string will now be properly passed to FILEDATA.

P1407 CONTROLLER - "PD" EQUAL VS. NONRESIDENT FILES

The SPO request PD= or PD <directory name>/= will now display "NULL FILE" kind for all non-resident files. This is due to the fact that the FILEKIND of a non-resident file cannot be guaranteed as different versions of the same file title can be of different file kinds.

The FILEKIND of any non-resident file and all its versions can be found by entering PD <filename>.

DATA COMMUNICATIONS

D2477 DATACOM - DATA COMM SUBSYSTEM ENHANCEMENT

The Data Comm Subsystem has been enhanced. See the attached User's Guide for details of implementation.

D2545 DATACOM - "DCRECON" NO LONGER VISIBLE

DCRECON is no longer a visible stack. DCRECON will now only appear on the mix page when it is awaiting an RSVP. In addition, DCRECON will no longer be scheduled, thereby speeding the exit of job awaiting DCRECON to clear datacom output prior to completion of a DS.

DATA COMMUNICATIONS

P1492 DATACOM - PREVENT "DCSTATUS" DUMP WITH EXTENDED TOGS

DCSTATUS was faulting with an INVALID INDEX due to an error in the tables returned to it by the MCP when extended toggles and tallies were used. This problem has been corrected.

DCALGOL

P1651 DCALGOL - LEFT TYPE TRANSFER FUNCTION

The left type transfer function will now allow MSGARRYIDS and MSGID.

Example:

```
BOOLEAN(MESSAGEID[3]):=TRUE;
```

DCP PROGRAM GENERATOR

P1009 DCPPROGEN - IMPROVED SWITCHED LINE HANDLING

The DCP was occasionally ignoring loss of carrier conditions when "LOSSOFCARRIER=DISCONNECT" was specified. This could cause a disconnection to be ignored, and data terminal ready (CD) left true. This problem has been corrected.

P1143 DCPPROGEN - "LINE INTERROGATE"

The LINE INTERROGATE DCWRITE function was not working properly for full-duplex lines. This problem has been corrected.

P1187 DCPPROGEN - FULL DUPLEX STATION NOT BUSY

If a station on a multi-drop full duplex line went not ready (by MCS control or terminate error), the entire line could be suspended indefinitely. This problem has been corrected.

P1360 DCPPROGEN - CORRECT ADDRESS USED AS A BYTEVARIABLE

When accessing the station address characters with the NDL bytevariable ADDRESS, DCPPROGEN was emitting code which incorrectly fetched the transmit address when the receive address was requested and vice versa.

Examples:

```
CHAR=ADDRESS(TRANSMIT)[1]
```

Was fetching byte 1 of the receive address

```
CHAR=ADDRESS(RECEIVE)[1]
```

Was fetching byte 1 of the transmit address.

This problem has been corrected; now, the proper receive or transmit address characters are accessed.

P8993 DCPPROGEN - "DCP" FAULT REPORT

The MCP could overlook a reported DCP fault and show that the DCP was off-line instead. The problem has been corrected.

P8999 DCPPROGEN - "INITIALIZE RETRY" WITH "LOCAL TABLES"

Execution of "INITIALIZE RETRY" in a line control routine would fail if the LOCALTABLES option were used. This problem has been corrected.

P9050 DCPPROGEN - "TERMINATE BLOCK"

Performing TERMINATE BLOCK in receive request could cause an END OF BUFFER condition on a subsequent STORE operation, even if a GETSPACE had been performed. This condition was occurring in the READPAPERTAPE request set. The problem has been corrected.

P9266 DCPPROGEN - CORRECT FETCH STORED TEXT

Using the FETCH statement to examine stored text on a switched line could cause the text area of the input message to be altered. This problem has been corrected.

DCP TEST GENERATOR

P1094 DCPTESTGEN - "$NEWSEQERR" OPTION

DCPTESTGEN will no longer get an INVALID INDEX fault when handling the NEWSEQERR option.

P1108 DCPTESTGEN - REQUIRE GOOD PARITY ON DISK

DCPTESTGEN will now terminate abnormally upon encountering an unrecoverable parity error when reading a temporary disk work file. Previously, such a parity error would cause DCPTESTGEN to go immediately to normal end-of-job.

DCSTATUS

P1141 DCSTATUS - "DCPANALYSIS" OUTPUT REFORMATTED

The DCPANALYSIS part of DCSTATUS was getting a SEG ARRAY fault at sequence number 01452000 when it attempted to analyze four or more DCP's. This problem has been corrected by printing the analysis for each DCP (or pair of DCP's, if exchanged) on a separate line. The formatting of remote output has also been improved in several instances.

P1142 DCSTATUS - "TERMINAL MAXINPUT/MAXOUTPUT" INFORMATION

DCSTATUS was reporting the value of MAXOUTPUT AS "BUFFER". It has now been changed such that the value of MAXINPUT/BUFFER is reported as "MAXINPUT" and MAXOUTPUT AS "MAXOUTPUT" when providing TERMINAL information.

DIAGNOSTIC MCS

P1578 DIAGNOSTMCS - INVALID CASE INDEX IN "STASTATECHNGR"

The procedure STASTATECHNGR, which is called to handle the "READY" command (among others), will now give a "MISSING <STATION I.D.>" error message if "ALL" is specified as the <station id>. Formerly, DIAGNOSTICMCS terminated with an INVALID INDEX fault.

DMS II - GENERAL

D2519 DMSII - USE OF "ADDRESSCHECK, CHECKSUM"

The following system note documents an existing feature which has not been previously documented.

ADDRESSCHECK is a DMSII feature which is invoked for a data base using the DASDL OPTION statement. Use of the ADDRESSCHECK option is very strongly recommended, because it provides extremely inexpensive protection from certain types of hardware failures. ADDRESSCHECK causes one extra word to be added to the end of each block of all data base data files. This extra word contains the block address of the block. When a block is read by the ACCESSROUTINES, the addresscheck word is checked to see if it equals the block address. If it does not equal the block address, some sort of data corruption has occurred (such as an undetected frame shift in the middle of reading the block). A fatal integrity error is then given by the ACCESSROUTINES.

CHECKSUM is a physical option, which means it can be set on a structure-by-structure basis. It can also be set as a global DASDL default using the DEFAULTS statement. CHECKSUM causes an extra word to be added to each block of a structure for which it is set (in addition to the addresscheck word, if any). The CHECKSUM is checked when a block is read from disk by the ACCESSROUTINES, and is recomputed before writing the block back to disk (if the block is updated). The CHECKSUM makes the probability of detecting any data corruption in the block extremely high; therefore, its use is strongly recommended.

Detecting data corruption when it occurs is vital, for if it is not detected until later, any rebuild or reconstruct using the DMSII audit trail will restore the corruption. For this reason, the ideal way to run is with both ADDRESSCHECK set and with CHECKSUM set for each structure. (CHECKSUM may not detect an error if the hardware starts a read at an incorrect segment address which happens to start on a block boundary.)

CHECKSUM may conveniently be added to an existing data base using the DMSII REORGANIZATION package, since it can be added to one or a few structures at a time. Adding ADDRESSCHECK is less convenient, because it applies to all structure or none; therefore, it cannot be added to an existing data base by the REORGANIZATION package. The data must be transferred to another data base (of otherwise identical descrption) in which ADDRESSCHECK is set.

DMS II - ACCESSROUTINES

D2244 ACR - LIMIT ERROR

A limit error on unordered and random data sets was "permanent"; i.e., even if a DASDL update operation increased the number of rows, the limit error persisted. Now, the ACCESSROUTINES notice that the file has been enlarged, and the limit error condition is corrected.

D2277 ACR - "SEGDESCABOVE"

The $ option SEGDESCABOVE has been employed to ensure that all D1 items which must be referenced by IRW's can be accessed for very large data bases.

D2278 ACR - DISPLAY STRUCTURE NUMBER

The display produced during compilation of ACCESSROUTINES (when ACRDISPLAY $ option is set) now includes the structure number as well as structure name.

D2279 ACR - ELIMINATE WARNING IF NO LOGICAL DATA BASES

A warning is no longer emitted when compiling the ACCESSROUTINES if the data base has no logical data bases.

D2288 ACR - PATH FIXING CONSISTENT

When the record referenced by a program's path is deleted (by that program or any other), its path must be adjusted. This path adjusting cannot be perfect without incurring excessive overhead; therefore, the following definition is made for structures with a logical ordering (index-sequential, ordered list and ordered data sets):

FIND NEXT will produce successive records in the collating sequence of the index, although some records added after the record was deleted may be skipped.

FIND PRIOR will produce all records in the collating sequence of the index prior to the deleted record, although some records may be retrieved which are later in the collating sequence than the deleted record.

Previously, the above definition held in most but not all cases. Now it holds in all cases.

The above may be clarified by an example. Consider an index sequential set S with keys A, K and Z stored.

1. Consider a program with a path to the set pointing at K. FIND S would retrieve the record with key K ("record K"). FIND NEXT S would retrieve record Z. FIND PRIOR S would retrieve record A.

2. Suppose that K is now deleted (by another program). FIND S would return a NOT FOUND exception. FIND NEXT S would retrieve record Z. FIND PRIOR S would retrieve record A.

3. Suppose that records B through Y are now added (by another program). FIND S would return a NOT FOUND exception as before. FIND NEXT S would retrieve Z ("skipping" L through Y). FIND PRIOR S would retrieve record Y (even though Y is later in the logical ordering than K was).

NOTE 1: A LOCK/STORE in which the key is changed is considered as a DELETE followed by a CREATE/STORE as far as all paths are concerned, including the set paths of the program doing the LOCK/STORE. Thus, for example, the program doing the LOCK/STORE and changing the key will always get a NOT FOUND exception on "FIND S" afterwards. Also, "FIND NEXT S" and "FIND PRIOR S" will normally return different results than they would if the set path of the program doing the LOCK/STORE were to track the record when the key changed. This may be clarified by an example. Consider an index sequential set S with keys A, K and Z stored. If the program's path for S references K and the key K is changed to L, then FIND S gives NOT FOUND, FIND NEXT S gives L (the record just STORED), and FIND PRIOR S gives A. If, however, the key K were changed to J, FIND S would give NOT FOUND, FIND NEXT S would give Z, and FIND PRIOR S would give J (the record just STORED).

NOTE 2: Ordered data sets will not be brought into full compliance prior to III.0 for reasons of audit file compatibility.

D2291 ACR - USE OF DESCRIPTION FILE

A new node containing copies of structure properties has been added to the description file in order to reduce the time needed to compile the ACCESSROUTINES and RECOVERY and to perform various functions using it.

D2438 ACR - BUFFER CORE IMPROVEMENT

The allocation of buffers by the ACCESSROUTINES has been improved for III.0. The DASDL specification for buffers has remained essentially the same except for some extensions for read ahead (see ACR note D2462); however, the ACCESSROUTINES now interprets it in a different manner.

Pre-III.0 Implementation:

For each file in the data base, the ACCESSROUTINES calculated a parameter called "XTRA" from the specification, as follows:

```
XTRA:=N+M*(number of users);
```

where N and M were specified in DASDL as:

```
D DATASET (
      .
      .
      .
  ) BUFFERS=N+M PER USER;
```

The XTRA parameter was adjusted when user programs would open and close the structure. (OPEN's incremented XTRA; CLOSE's decremented XTRA.) With respect to buffer allocation, the ACCESSROUTINES attempted to keep XTRA number of extra "not currently used" buffers in core. These extra buffers were in use at one time and were kept available in hopes that they would soon be reused.

Example:

```
D DATASET (
      .
      .
      .
  ) BUFFERS=1+1 PER USER;
```

With one user of structure D, 3 buffers would be allocated (2 XTRA plus 1 in use).

With 2 users of structure D, 5 buffers would be normally allocated (3 XTRA plus 1 in use for each user).

III.0 Implementation with READAHEAD off:

On III.0, the concept of XTRA has been replaced by "DECLAREDBUFFERS". When READAHEAD is FALSE, this parameter's calculation is identical to the former calculation of XTRA; however, the ACCESSROUTINES now attempt to keep at most DECLAREDBUFFERS in core.

Example:

```
D DATASET (
      .
      .
      .
  ) BUFFERS=1+1 PER USER;      % READAHEAD = FALSE
```

With one user of structure D, only 2 buffers (formerly 3) will be allocated.

With 2 users of structure D, only 3 buffers (formerly 5) will be allocated.

Following is an example of II.9 DASDL specification and III.0 extended DASDL specification. Either specification has the same effect in a non-READAHEAD environment.

II.9 Example

```
BUFFERS=M+N PER USER
```

III.0 Example

```
BUFFERS=M+N PER RANDOM USER . . .
```

III.0 Implementation with READAHEAD on:

Consider the following buffer specification for READAHEAD:

```
BUFFERS=M+N PER RANDOM USER
       OR P PER SERIAL USER . . .
```

The DECLAREDBUFFERS parameter has the same semantics as above, but is calculated slightly differently, as follows:

```
DECLAREDBUFFERS=M+N*(no. of random users)
                +P*(no. of serial users).
```

Clearly, the only difference in this case is that the "P" parameter is substituted for the "N" parameter for all serial users.

Note: On both the pre-III.0 and III.0 implementations, the actual number of buffers to be allocated is not completely constrained by the user specification. One obvious case is a user specification of BUFFERS=0+0. (The ACCESSROUTINES must allocate at least 1 buffer, or the program will not run.) Moreover, the ACCESSROUTINES may exceed the user specification in order to employ certain audit algorithms, table splitting, etc.

The III.0 implementation is a more reasonable approach to buffer management for the following reasons:

1. The ACCESSROUTINES will more precisely allocate buffers according to what the user specified. This increases the user's control and predictability of buffer core.

2. The former implementation used too many buffers for most applications, effectively decreasing total system throughput without increasing the performance of the data base. (The default buffer specification of 1+1 PER USER (without READAHEAD) is ample for most applications.)

3. The III.0 algorithm keeps the amount of buffer core lower; hence, data base total core may be postponed from reaching ALLOWEDCORE. This is of tremendous benefit, since overlaying and thrashing occur when ALLOWEDCORE is exceeded.

Note: Data bases that have been finely tuned under the old algorithm will use less core, but may sometimes do excessive I/O under the new algorithm. If so, the number of buffers per structure may have to be increased for some structures, especially index-sequential.

D2462 ACR - READ AHEAD BUFFERING

INTRODUCTION

A user program can access a data base structure in one of two modes, as follows:

1. Random

   Successive data management operations access non-adjacent blocks of the structure from one call to the next.

   Example:

   ```
   FIND <structure> AT . . .;
   ```

   Repeated operations of this type against a RANDOM <structure> would normally yield random access.

2. Serial

   Successive data management operations access the same or "logically adjacent" blocks of the structure. Logically adjacent blocks are blocks which would be hit by repeated "FIND NEXT <structure>" selections one after the other. For certain structures, this implies that the blocks need not be physically adjacent (e.g., INDEX SEQUENTIAL). This type of access may occur on any of the following statements:

   ```
   A. FIND NEXT <structure>;
   B. FIND NEXT <structure> AT . . .;
   C. FIND FIRST <structure> AT . . .;
   D. FIND <structure> AT . . .;
   ```

When a user program accesses a structure serially, it is possible for the ACCESSROUTINES to issue reads on blocks in advance of when they are currently required. This process is known as "read ahead". The benefit of read ahead is a decrease in buffer wait times when blocks are actually required. This in turn causes a decrease in total elapsed time for serial accessing programs.

Read ahead has been implemented for all types of physical structures (data sets and index sets) in the "next" direction only (not "prior"). The remainder of this document describes how this feature may be invoked.

READ AHEAD CRITERIA

Read ahead will be performed on a user/structure basis when the following criteria are met:

1. DASDL has been extended to allow the following:

   a. The read ahead option may be turned on or off for any physical structure.

   b. The number of buffers per read ahead use may be specified.

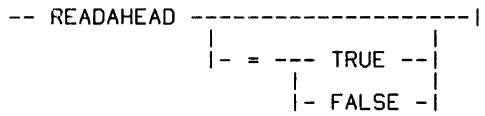
      The ACCESSROUTINES will only attempt read ahead for a user if the read ahead option has been turned on and the number of buffers per read ahead user is at least 2.

2. The user program must be accessing the structure (with read ahead enabled) in a serial manner.

DASDL SPECIFICATIONS

A new physcal attribute, "READAHEAD", has been implemented for data sets and index sets and may not be specified for accesses.

Syntax:

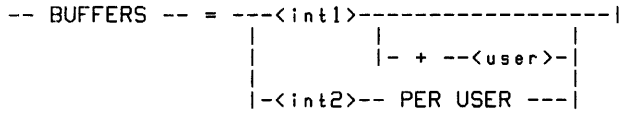


READAHEAD is FALSE by default.

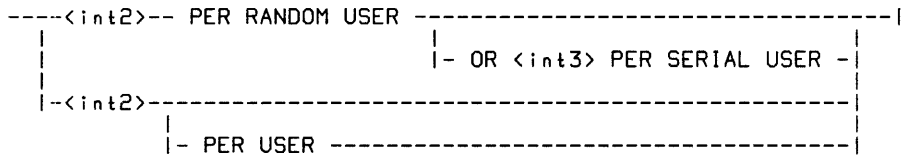
The purpose of this attribute is to inform the ACCESSROUTINES whether or not to perform read ahead when a user program accesses the structure serially.

Also, the BUFFERS clause has been extended.

Syntax:



<user>



This buffer syntax is compatible with the former syntax, and the parameters <int1> and <int2> have retained their original default values; however, the semantics have changed slightly.

1. <int1>

   This is the number of base buffers allocated for the structure.

2. <int2>

   The semantics is the same for all occurrences of <int2> in the syntax. This is the number of buffers allocated to each user accessing the structure randomly. If READAHEAD is set to FALSE, all users are assumed to be random.

3. <int3>

   This is the number of buffers allocated to each serial user of the structure. When READAHEAD is TRUE, the value of this parameter must be greater than or equal to 2; otherwise, a syntax error is generated. This restriction helps to ensure that the ACCESSROUTINES will have allocated sufficient buffers to handle serial users without thrashing. If READAHEAD is FALSE, this parameter is ignored.

The default settings for these parameters are as follows:

   A. <int1> - 1
   B. <int2> - 1
   C. <int3> - Maximum of 2 or <int2>+1

To facilitate a II.9 to III.0 DASDL update, both the READAHEAD and BUFFERS clauses may be specified in the DEFAULTS clause for "global", data set or set defaults.

HOST LANGUAGE STATEMENTS CAUSING READ AHEAD

If READAHEAD was specified in DASDL, the ACCESSROUTINES will perform read aheads where applicable. No special host language command(s) is necessary to force the program into a serial mode; however, read ahead may only occur on the following verbs:

1. FIND
2. LOCK(MODIFY)
3. DELETE

There are several variations of read ahead, depending on whether selections are made via the data set or index set.

Consider the following DASDL:

```
DEFAULTS (READAHEAD=TRUE);
D DATASET (
            .
            .
            .
          );
S SET OF D . . .;
```

Repeated execution of the following selections (shown with the "FIND" verb) may yield read aheads for D or S:

1. FIND NEXT D;

   Read aheads will be performed for D but not S.

2. FIND KEY OF NEXT S;

   Read aheads will be performed for S but not D.

3. FIND KEY OF S AT . . .;
   FIND KEY OF FIRST S AT . . .;
   FIND KEY OF NEXT S AT . . .;

   If logically consecutive blocks of S are being hit by any of the above selections, read aheads will be performed for S. No read aheads will be performed for D.

4. FIND NEXT S;

   Read aheads will be performed for both S and D.

5. FIND S AT . . .;
   FIND FIRST S AT . . .;
   FIND NEXT S AT . . .;

   If logically consecutive blocks of S are being hit by any of the above selections, read aheads will be performed for S. The same is true for D.

Note: If READAHEAD is off for a structure, read aheads will never be performed for that structure.

ACCESSROUTINES READ AHEAD ALGORITHM

When a user program performs one of the data management operations shown above, the ACCESSROUTINES execute read ahead detection code. This code compares the block pointer of the previous access to the block pointer of the current access. If the two accesses occurred in the same or logically adjacent blocks, it is counted as a sequential access. After 3 such consecutive accesses, the program enters serial mode. This also causes the buffer pool to be adjusted according to the BUFFERS clause in DASDL. Once serial mode is entered, one read ahead is performed each time another block is accessed sequentially. If sequential access is discontinued, read aheads are no longer issued and random accesses are counted. If 6 consecutive random accesses occur, the program will revert to a random state. The buffer pool will be readjusted and read aheads will no longer occur. If the program changes state more than once (random to serial, serial to random), the number of accesses needed to cause a state change will be raised to a higher value to make state changes much more difficult. This will prevent spurious read aheads and buffer allocations and deallocations.

DMS II - ACCESSROUTINES

P1017 ACR - "ACCESSROUTINES" SEGMENTATION

Some related procedures and blocks have been combined into the same code segment in order to decrease the number of segment descriptors in the data base stack.

P1018 ACR - IMPROVE "I/O" ERROR DISPLAY

The I/O error messages emitted by the ACCESSROUTINES have been improved.

P1019 ACR - ORDERED DATA SETS OUT OF SEQUENCE

A problem which could result in the records of an ordered data set being out of sequence has been corrected.

P1020 ACR - CONTROL FILE LOCKING

The ACCESSROUTINES no longer access the control file without acquiring the appropriate lock(s). Formerly, I/O errors or reading invalid records from the control file may have occurred.

P1057 ACR - AUDITED BIT VECTOR "ZERO DISK ADDRESS"

Insert after a generate null was causing a data base fatal "ZERO DISK ADDRESS" error. This has been corrected.

P1058 ACR - "SEG ARRAY" FAULT WHEN CREATING PARTITIONS

It was possible to get a SEG ARRAY fault when creating partitions with large titles (an internal array was too small). This problem has been corrected.

P1071 ACR - "DMSCLOSEERROR 2" AT END OF HALT/LOAD RECOVERY

If a global data population item were specified for the restart data set, it was impossible to successfully complete Halt/Load recovery. Instead, an INVALID INDEX occurred, followed by DMSCLOSEERROR 2. This has been corrected.

P1153 ACR - STORAGE CLOSE IN UNAUDITED DATA BASES

If an unaudited data base has been updated, but the last user program to close the data base is not an updater (e.g., OPEN INQUIRY), the procedure which writes the storage variables back to the file (STORAGEOPENECLOSE) is not called. This error could have serious consequences; it has been corrected. This problem did not exist prior to II.9, nor does it exist for audited data bases.

P1199 ACR - MARK STACK "DS" IF ABORT DIES

Under some conditions, abort recovery would fail, but the stack which initiated it would not be marked as DS'ed. This has been corrected.

P1200 ACR - "SEG ARRAY" IF PACK AUDIT

The ACCESSROUTINES would get a SEG ARRAY error at 10274000 if auditing to system resource pack and copy to tape or verify were specified. This no longer occurs.

P1274 ACR - MISSING "HANDLEWRITEERROR" CALL

Under certain conditions, write errors would not be noticed by the ACCESSROUTINES; therefore, the row might not be locked out, and the BLKIMG audit record would not be produced. This problem has been corrected.

P1342 ACR - INNOCENT PROGRAMS HANG ON DISEASED "DBS"

It is now less likely that programs will hang on a data base that is hung but which they are not using. (This could happen at EOT when DMSFREE is called to free all the records of this stack in all currently open data bases.)

P1344 ACR - "SET ROWLOCK" READ, WRITE ERRORS

The ACCESSROUTINES were not setting the ROWLOCK field to 2 if a read error were encountered on the row. For unaudited data bases, the ACCESSROUTINES were not setting the ROWLOCK field to 1 if a write error were encountered on the row.

P1345 ACR - RETRY DISK READ ON CHECKSUM ERROR

The ACCESSROUTINES will now reread the disk block if a checksum error is encountered but no read error is reported by the MCP. They will repeat this several times, if necessary, before giving up and reporting an I/O error.

P1346 ACR - ABORT DUE TO "BAD DIVEST"

Space allocation for ORDERED DATA SETS, under a certain condition, could cause the data base to become DSed by a BAD DIVEST. This has been corrected.

P1400 ACR - FALSE ERROREXIT "9" IN "DIVEST" UNDER "DEBUG"

If the compile-time option DEBUG (which is used for internal debugging purposes) were set when compiling the ACCESSROUTINES, it was possible to get an erroneous errorexit 9 (DIVEST ERROR) if multiprocessing and overlay were occurring. This has been corrected.

P1469 ACR - ALL ZERO BLOCKS/RECORDS

When ALLOWEDCORE is set low enough, OVERLAY will deallocate buffers. There was a timing problem, such that an overlaid buffer could be written as all zeroes. This problem has been corrected.

P1471 ACR - ATTRIBUTE ERRORS ON CONTROL FILE

If a DM OPEN ERROR exception occurred when two or more programs opened a data base simultaneously, file attribute errors could have been generated causing an error condition in the control file module. This has been corrected.

P1521 ACR - RECOVERY OF PARTITIONED STRUCTURES

Several problems involved with aborted recovery of partitioned structures have been corrected. These problems were associated with the following:

1. Backout of original partition file creation.

2. Recreation of partition file.

3. An abort which left the end of file for a partition at the end of its last row.

4. A second abort of partition changes immediately after a prior abort on the same partition.

P1532 ACR - "UNIDENTIFIED MISCELLANEOUS ERROR" RECOVERY

The system no longer gets at "UNIDENTIFIED MISCELLANEOUS ERROR" in RECOVERY at 50329820 due to an unexpected restart type of 3 or 4 in a last good restart area audit record. When last good restart areas are stored by CLOSE in the case of abnormal program termination, the STORE is coordinated with any ABORT recovery that may be happening concurrently. Duplicate restart areas are no longer stored in the case that a Halt/Load or abort recovery happened soon after storing such a restart area.

P1533 ACR - "ZERO FIND ADDRESS" AFTER "INSERT"

The ACCESSROUTINES no longer terminate with "ZERO FIND ADDRESS" when INSERT <set> is followed by FIND NEXT <set>.

P1587 ACR - SERIAL ACCESS OF INVALID STATE RECORDS

There are three states for standard variable format records: normal, deleted and invalid. The invalid state exists only after a CREATE-STORE has failed to store a record after having allocated a new record slot. If serial access (e.g., FIND NEXT D) encounters that slot, it fails to compute its length properly and probably faults. Deleted records do not cause this problem.

P1594 ACR - DEADLOCK DURING ON-LINE RECONSTRUCT

It was possible for a deadlock to occur during on-line reconstruction if any user programs locked records while in transaction state. This has been corrected.

P1613 ACR - FIX AUDIT LASTRECORD; ZIP "COPYAUDIT"

If both update and inquiry programs had the data base open and the last program to close the data base were an inquiry program, the audit file would be incorrectly terminated with SDSEOF records and the last record attribute was incorrect. These problems have been corrected.

When the ACCESSROUTINES decline to extend an existing disk or pack audit file, COPYAUDIT is zipped for it, if appropriate.

P1616 ACR - ERROR IN "SET TO ENDING" ORDERED DATA SETS

A problem has been corrected in which the ACCESSROUTINES would not find "PRIOR" after a "SET TO ENDING" against an ordered data set when records were present.

P1678 ACR - ERRORS IN RECOVERY CLOSE

The ACCESSROUTINES will now terminate if an error is detected when RECOVERY closes the data base. Formerly, errors were ignored and unpredictable results could occur.

It should be noted that since the errors were not handled in close RECOVERY, a subsequent ABORT could hang the data base.

P1679 ACR - DEADLOCK RECONSTRUCTION VS. SYNCPOINT

A timing window has been closed in which reconstruct would hang if the final "stop phase" of the data base occurred during the middle of a syncpoint.

P1680 ACR - RESULT ON "SELECT" REJECT

The ACCESSROUTINES no longer fail to generate a proper DMSTATUS result for INQUIRY when a record does not pass the logical data base SELECT condition.

P1681 ACR - EMBEDDED DATA SET PATHS NOT INITIALIZED

When a FIND or LOCK was executed on the workarea of the master, the current structures of embedded data sets below it were not initialized to "BEGINNING". A possible consequence was that a subsequent "DELETE <embedded data set name>" would typically get a fatal data base error, such as "MISSING ENTRY FROM AUTO SET/SUBSET". Now, embedded structures are always set to "BEGINNING" when a LOCK or FIND is done on a master (even if the same master is found).

P1682 ACR - INSERT/REMOVE RESET BITVECTOR

ACCESSROUTINES patch 29.77 caused bitvectors to be reset to the beginning when an INSERT or REMOVE was done on them. This has been corrected.

P1683 ACR - ATTRIBUTE ERROR "RSFILE.IOCLOCKS"

If statistics were set and the last program to close the data base was an inquiry program after one or more update programs previously closed the data base, a harmless attribute error at 92089020 in the ACCESSROUTINES occurred, because RSFILES was erroneously closed by the last updater. Also, unpredictable actions could occur, because two processes were using a FIB without the protection of a lock. The errors no longer occur.

P1684 ACR - HALT LOAD RECOVERY USES WRONG AUDIT FILE

If on-line data recovery was running, it would reset the audit file number in the control file every time it created a restart point and when it finished. It would reset it to the audit file current when it was initiated. Normally, the ACCESSROUTINES would correct it at the next audit file switch or when the last update closed the data base; however, if a halt/load occurred before it was corrected, RECOVERY would use the wrong audit file and corrupt the data base as well as destroy the continuity of the audit. The error has been corrected.

P8946 ACR - INTERLOCK OF ABORT AND RECONSTRUCTION

If reconstruction tried to stop the data base while abort was releasing the data base, abort could cause the data base to be released while reconstruction thought it was successfully stopping it. This synchronization problem has been corrected.

P9013 ACR - RETURN CORRECT STRUCTURE NUMBER

The ACCESSROUTINES now returns the correct structure number on an exception for remaps.

P9018 ACR - "I/O" ERRORS ON OPEN

A fatal error will now result if an I/O error occurs while trying to open a data base file; previously, the program would try to proceed and cause other strange errors.

P9020 ACR - "INVALID INDEX," VARIABLE FORMAT

If the creation of a variable format record failed (e.g., due to duplicates exception), it was possible for a subsequent FIND NEXT operation on the data set to fail with INVALID INDEX in the GETDATA procedure. This has been corrected.

P9073 ACR - DELETE EMBEDDED DATASET ERROR

Depending on the particular DASDL, it was possible to get an error when deleting master records which formerly contained detail records in an unordered data set. This has been corrected.

P9084 ACR - INVALID "DATAERROR 4"

An invalid DATAERROR 4 no longer occurs when attempting to store a variable format record. Formerly, the ACCESSROUTINES would emit this error if the CREATE were performed with a normalized real as the record type (e.g., CREATE D (R);, where R is a normalized real would yield at DATAERROR 4 at store time).

P9094 ACR - FAIL WHEN PRINT STATISTICS

If the restart data set were never invoked and opened, the ACCESSROUTINES would fail when the statistics were printed. DMCLOSEERROR #2 would result, and sometimes a dump by FORGETCHECK also. The problem has been corrected.

P9096 ACR - INCORRECT RESTART RECORD

It was possible for the wrong restart record to be stored in the restart data set if a program were abnormally terminated outside of transaction state after another program had caused an abort. In this case, the saved restart area was from before the abort, not after. This has been corrected.

P9177 ACR - INITIALIZATION OF VARIABLE-FORMAT PARTITIONS

If a partition of a standard variable format data set were initialized after using another partition, it was possible for the new partition to "inherit" the available address table (DKTABLE) from the previous partition. This could cause serious errors in the new partition, including corrupted data and fatal data base errors. This has been corrected.

P9178 ACR - STANDARD VARIABLE FORMAT AVAILABLE TABLES

The ACCESSROUTINES no longer lose available space tables for standard variable format data sets. Formerly, these tables were never written beyond the original data area, and thus were completely forgotten. Now, available space tables are written past the data area, if there are no empty blocks in the data area. (Data is still never written past the data area).

P9181 ACR - CORRECT ORDERED SET PATHS

User paths were not adjusted properly when ABORT backed out a create of the last record in an ordered data set block. This has been corrected.

P9185 ACR - ELIMINATION OF HOLES IN RESTART DATA SET

The ACCESSROUTINES no longer use records in the restart data set as temporary storage for last good restart areas. This results in less processor and I/O overhead, eliminates a potential source of fatal I/O errors in RECOVERY, simplifies the DMSII software, and can result in slightly less data being written to the audit trail.

It is very likely that rebuild would fail at sequence number 60933000 in RECOVERY. This has been corrected.

P9251 ACR - TOO MANY TABLE LEVELS CAUSE "INVALID OP"

The ACCESSROUTINES no longer cause an INVALID OP when attempting to allocate more levels of index requesting tables than the software can handle. This condition is now detected, an appropriate message is displayed, and the data base is DS'ed.

P9256 ACR - CORRECT PATH FOR DISJOINT ORDERED DATA SETS

If an ordered data set path or ordered access path had extended beyond the last record in the data set, an abort occurred. A FIND NEXT after the abort could have found a record instead of getting a NOT FOUND. This has been corrected.

P9257 ACR - DATA CORRUPTION IN ROOT WORD

A root word is a link to the root of an embedded structure which resides in a master record of the next higher structure. For several types of embedded structures (index sequential, unordered data sets, etc.), bits 19:20 of the root word should be zero, unless all bits of the root word are ones.

Violation of this condition (which are instances of data corruption) are now detected, a message is displayed, and these bits are zeroed so that the data base can keep running.

If the user program which detects a violation has the data base open update, the root word will also be corrected on disk in the data base and an appropriate message displayed. The user should take note of these messages, because if a root word is corrupted, there may very well be other corrupted data in the data base. The sooner corrective action is taken, the easier it will be to correct the situation.

P9260 ACR - ACCELERATE NOT FOUND LINEAR SEARCH

Linear search for index sequential and ordered list will now return a NOT FOUND without going to the end of the file if the high order portion of the key has been specified.

P9261 ACR - FIND NEXT DIRECT AT KEY > VALUE FAILS

When the current path of the access A to a direct data set was pointing to a record with key value X, "FIND NEXT A AT KEYNAME > X" would return a NOTFOUND exception instead of finding record X+1. This problem has been corrected.

Other errors have also been corrected for FIND's via a direct access, including selection expressions involving zero.

P9264 ACR - ELIMINATE "MASKSEARCH" IN "FIND KEY"

"FIND KEY OF" has been made more efficient by eliminating a MASKSEARCH.

P9280 ACR - PARTITIONED INDEX SETS

It is sometimes necessary for the system to "extract" a key value from a data record, and use that value to look up an entry in an index set; e.g., DELETE FIRST <data set>, when there are one or more sets for that <data set>. If any such set happens to be partitioned, and if this is the first use of that set for this partition master, the partition code interferes with the DELETE, and errors will occur. Such errors include not finding the entry in the set, which is a fatal data base error.

P9281 ACR - ORDERED ACCESS PATH

1. A FIND NEXT on an empty data set now sets the path to ENDING instead of BEGINNING.
2. A FIND NEXT AT KEY > VALUE, which gets a NOTFOUND, now sets the path to ENDING instead of BEGINNING.

P9283 ACR - WARNINGS SUPPRESSED

Warnings of empty node lists are no longer generated for data sets or remaps which have a record type declared but no variable format parts.

P9284 ACR - COMPILE-TIME PRINTING

Redundant information about remaps is no longer printed when the ACRINFO option is set.

DMS II - BDMSALGOL

D2488 BDMSALGOL - "OPEN TEMPORARY" AND "CLOSE LOCK"

It is now considered an error to open a data base with the TEMPORARY option and to close a data base with a LOCK option.

The compiler will flag as errors all data base OPEN statements that specify the TEMPORARY option and all data base CLOSE statements that specify the LOCK option.

Programs that had specified either an OPEN TEMPORARY or a CLOSE LOCK and were compiled with II.9 or earlier compilers will not be allowed to run on III.0 ACCESSROUTINES. A run-time error will occur in this case.

DMS II - BDMSALGOL

P1280 BDMSALGOL - BINDING "DMSII" ACCESSES

It is now possible to bind a data base that declares an access. Previously, no information was saved for an access; thus, incorrect binding information was created.

P1486 BDMSALGOL - ALLOW "-" IN DATA BASE IDENTIFIER

If the data base name or logical data base name in a data base declaration contained a hyphen "-", the compiler would not handle the name correctly.

Example:

```
DATABASE TESTDB-L1;
DATABASE LDB-NAME OF DB;
```

Hyphens are now recognized correctly in data base identifiers.

DMS II - BDMSCOBOL

D2488 BDMSCOBOL - "OPEN TEMPORARY" AND "CLOSE LOCK"

It is now considered an error to open a data base with the TEMPORARY option and to close a data base with a LOCK option.

The compiler will flag as errors all data base OPEN statements that specify the TEMPORARY option and all data base CLOSE statements that specify the LOCK option.

Programs that had specified either an OPEN TEMPORARY or a CLOSE LOCK and were compiled with II.9 or earlier compilers will not be allowed to run on III.0 ACCESSROUTINES. A run-time error will occur in this case.

DMS II - BDMSCOBOL

P1283 BDMSCOBOL - COMPARES INVOLVING DATA SET NAMES

Alphanumeric compares involving data set names are now allowed. Previously, an incorrect error message was produced for a valid comparison.

P1284 BDMSCOBOL - DATA BASE SECTION SYNTAX

The compiler is no longer discontinued while attempting to compile invalid syntax in the data base section.

P1412 BDMSCOBOL - GLOBAL DATA BASES

The compiler now allows the global data associated with a data base to be referenced by a bound program unit declaring the data base as a global data base.

P9110 BDMSCOBOL - "XREF" "DM" KEYS IN "SELECTION" EXPRESSIONS

The COBOL compiler no longer fails to XREF DM keys in SELECTION expressions.

DMS II - BDMS/PL/I

D2488 BDMSPLI - "OPEN TEMPORARY" AND "CLOSE LOCK"

It is now considered an error to open a data base with the TEMPORARY option and to close a data base with a LOCK option.

The compiler will flag as errors all data base OPEN statements that specify the TEMPORARY option and all data base CLOSE statements that specify the LOCK option.

Programs that had specified either an OPEN TEMPORARY or a CLOSE LOCK and were compiled with II.9 or earlier compilers will not be allowed to run on III.0 ACCESSROUTINES. A run-time error will occur in this case.

DMS II - BDMS/PL/I

P1120 BDMSPLI - REUSE SCRATCH STACK AREAS IN KEY COMPARES

Under certain circumstances, the scratch string stack or scratch word stack was not being cut back during the key compare procedure for FIND statements. This would cause the string stack or word stack to grow unnecessarily, sometimes causing an INVALID INDEX or SEG ARRAY error to occur.

The string stack or word stack are now reused when possible during a key compare procedure.

P1280 BDMSPLI - BINDING "DMSII" ACCESSES

It is now possible to bind a data base that declares an access. Previously, no information was saved for an access; thus, incorrect binding information was created.

DMS II - BUILDINQUIRY

P1201 BUILDINQ - "INVALID INDEX"

An incorrect test failed to resize an array when necessary, which subsequently caused an INVALID INDEX. This has been corrected.

P1202 BUILDINQ - MISSING SETS

If an INQUIRY program were generated for a logical data base or selected data sets, it was possible to accidentally exclude some sets. This could result in faults or less than optimum search algorithms. This has been corrected.

P1535 BUILDINQ - "BUILDINQ" FAILS WITH LOGICAL DATA BASE

BUILDINQ could fail for the following situation: A logical data base contains a remap, which has a set with data in key. The nature of the error was unpredictable, but was not likely to allow the BUILDINQ program to finish.

P1595 BUILDINQ - LOOPING

BUILDINQ could loop when generating an inquiry program for a logical data base. This has been corrected.

P1685 BUILDINQ - BUILDING STRUCTURES NOT IN LOGICAL DATA BASE

BUILDINQ will now generate only global items and will drop data sets if global data is included in a logical data base.

DMS II - BUILDREORGANIZATION

P1534 BUILDREORG - "INVALID INDEX" WITH TOO MANY STRUCTURES

BUILDREORG no longer terminates with an INVALID INDEX when the structure numbers exceed 250.

DMS II - COPY AUDIT

P1586 COPYAUD-II - COPY PRIMARY AUDIT ON TAPE

COPYAUDIT would not copy a primary audit if it was on tape unless the OVERRIDE option were used, because of setting the attributes for NEXTTAPE improperly. This has been corrected.

DMS II - DASDL

D2411 DASDL - NEW DOLLAR OPTIONS

Several new dollar options have been implemented to more exactly control some of the output to the compiler file LINE. The output affected is that produced by the TEST and STORE options. These options will continue to function as they have in the past, but they may be deimplemented on a future release.

The new options are: TEXT, FILE, LAYOUT, DECKLIST, STRUCTURE.

TEXT: Will cause printing of the DASDL-generated source statements used in the ACCESSROUTINES (also produced by TEST).

FILE: Will cause printing of storage requirements and file size information (also produced by STORE).

LAYOUT: Will cause printing of item and key item offsets and sizes (also produced by TEST or STORE).

DECKLIST: Will cause printing of the deck which can be zipped to compile the database software: ACCESSROUTINES, RECOVERY, etc., (also produced by TEST).

STRUCTURE: Will cause printing of a new listing of the structures showing hierarchies via indentation and collecting data sets, set, access and remaps together according to their associations.

One other option, ALLINFO, is simply a shorthand for all five of the above options. SET, RESET or POP on ALLINFO is exactly the same action as on each of the above individually.

The following example shows DASDL input.

```
$SET ALLINFO
D-1 DATA SET
  (
    N1 NUMBER(10);
    N2 NUMBER(10);
    E-1 DATA SET
      (
        M1 NUMBER(10);
        M2 NUMBER(10);
      );
    SE-1 SET OF E-1 KEY M1;
    RE-1 REMAPS E-1
      (
        M1;
        M2;
      );
  );
R-1 REMAPS D-1
  (
    N1;
    N2;
    RE-1;
  );
D-2 DATA SET
  (
    X1 ALPHA(10);
    X2 ALPHA(15);
  );
```

The following shows the output produced by the DASDL input example.

```
                ========================================================
                THE FOLLOWING SECTION IS PRODUCED BY TEXT (OR TEST) OPTION
                ========================================================


                D-1 (2)
VALIDRECORD     ** (HB4[0] NEQ MAXA FOR 10 OR HB4[10] NEQ MAXA FOR 10)#
VERIFYSTORE     ** ((UW4[0] NEQ MAXA FOR 10 OR UW4[10] NEQ MAXA FOR 10))#
INVALID         ** REPLACE HB0[0] BY DEFAULTNULL FOR 3 WORDS;#
CLEARDATA       ** REPLACE HB0[0] BY DEFAULTNULL FOR 3 WORDS;#
DMSTOUSER       ** REPLACE UW8 BY HB8 FOR 10;#
USERTODMS        ** REPLACE HB8 BY UW8 FOR 10;#
USERTODMS       ** REPLACE HB8 BY UW8 FOR 10;#
```

```
                D-2 (7)
VALIDRECORD     ** (HB8 NEQ MAXA FOR 10 OR HB8[10] NEQ MAXA FOR 15)#
VERIFYSTORE     ** ((UW8 NEQ MAXA FOR 10 OR UW8[10] NEQ MAXA FOR 15))#
INVALID         ** REPLACE HB0[0] BY DEFAULTNULL FOR 5 WORDS;#
CLEARDATA       ** REPLACE HB0[0] BY DEFAULTNULL FOR 5 WORDS;#
DMSTOUSER       ** REPLACE UW8 BY HB8 FOR 25;#
USERTODMS       ** REPLACE HB8 BY UW8 FOR 25;#
USERTODMS       ** REPLACE HB8 BY UW8 FOR 25;#
```

<page eject>

==========
THE FOLLOWING SECTION IS PRODUCED BY FILE (OR STORE) OPTION
==========

```
 FILE : ""D-1"""/DATA."

    BLOCKSIZE  =      140 (RECORDS)
    RECORDSIZE =        3 (WORDS)
    AREASIZE   =     7840 (RECORDS)
    AREAS      =        2 (ROWS)
    POPULATION =    10000 (RECORDS)
  ON DISK
    BLOCKSIZE  =      420 (WORDS)
    AREASIZE   =      784 (SEGMENTS)
    MAX SPACE  =     1568 (SEGMENTS)

                         .
                         .
                         .

 FILE : ""D-1"""/"""E-1"""/"""SE-1""".."

    TABLESIZE  =       28 (KEY ENTRIES)
    AREASIZE   =      390 (TABLES)
    AREAS      =       26 (ROWS)
    POPULATION =       10 (MEMBERS)
    LOADFACTOR =       66%
  ON DISK
    BLOCKSIZE  =       56 (WORDS)
    AREASIZE   =      780 (SEGMENTS)
    MAX SPACE  =    20280 (SEGMENTS)


  TOTAL STORAGE REQUIREMENTS FOR DATA BASE:

                        35288 SEGMENTS ON HPT DISK
                            0 SEGMENTS ON DISK PACK
                            0 SEGMENTS ON DISK PACK
```

<page eject>

==========
THE FOLLOWING SECTION IS PRODUCED BY LAYOUT (OR TEST OR STORE) OPTION
==========

R E C O R D L A Y O U T I N F O R M A T I O N

RECORD AND KEY AREA OFFSETS ARE GIVEN AS DIGITS = (WORD,DIGIT)
(4 BIT DIGITS).
SIZES ARE GIVEN IN DIGITS.
NUMBERS AT THE LEFT INDICATE IN WHICH VARIABLE FORMAT PART THE
ITEM IS FOUND.
FOR SETS, "*" AT THE LEFT INDICATES THAT THE ITEM IS KEY DATA.

STRUCTURE 2: D-1, DATA SET

(RECORD TYPE)	ITEM NAME	OFFSET	SIZE
	N1	0 = (0, 0)	10
	N2	10 = (0,10)	10
	E-1	0 = (0, 0)	12
	SE-1	24 = (2, 0)	12
	RE-1	0 = (0, 0)	12

```
                    .
                    .
                    .
STRUCTURE 6: R-1, REMAP

     (RECORD TYPE)      ITEM NAME          OFFSET          SIZE

                        N1                  0 = (   0, 0)     10
                        N2                 10 = (   0,10)     10
                        RE-1                0 = (   0, 0)     12

                   <page eject>


                      ======================================================
                      THE FOLLOWING SECTION IS PRODUCED BY STRUCTURE OPTION
                      ======================================================


DATA SET: D-1 (2)
          REMAPS: R-1 (6)

          DATA SET: E-1 (3)
                    REMAPS: RE-1 (5)
                    SETS:   SE-1 (4)

DATA SET: D-2 (7)

                   <page eject>


                  ================================================================
                  THE FOLLOWING SECTION IS PRODUCED BY DECKLIST (OR TEST) OPTION
                  ================================================================


? JOB "CODE";
  CLASS = 40;
BEGIN
  COMPILE  ACCESSROUTINES/"CODE" WITH DMALGOL LIBRARY;
  COMPILER FILE TAPE = DATABASE/SYMBOLIC,
           RECOVERYSYM = DATABASE/RECOVERYSYMBOLIC,
           DASDL = DESCRIPTION/"CODE",
           PROPERTIES = DATABASE/PROPERTIES;
  DATA CARD
$ MERGE LINEINFO
$ SET ACRDISPLAY
? COMPILE  UTILITY/"CODE" WITH DMALGOL LIBRARY;
  OPTION = FAULT ARRAYS CODE BASE FILES DBS;
  COMPILER FILE TAPE = DATABASE/UTILITY,
           DASDL = DESCRIPTION/"CODE",
           PROPERTIES = DATABASE/PROPERTIES;
  DATA CARD
$ MERGE LINEINFO
? END JOB.
```

D2437 DASDL - "REORGANIZE GLOBAL DATA" EXTENDED

The syntax for the reorganization specifications for global data has been extended. The new syntax is as follows:

```
-- REORGANIZE GLOBAL DATA ------------------------------------ ; -----|
                          |                                          |
                          |- ( -- ITEMS --- SAME ------ ) -|         |
                                             |           |           |
                                             |- CHANGED -|           |
```

If neither ITEMS SAME nor ITEMS CHANGED is specified, it will be as if ITEMS CHANGED were specified. This action is compatible with previous releases, where no other option was available.

D2460 DASDL - MAINTAIN "UPDATE" LEVEL

The description file UPDATE level number will no longer be unconditionally increased on every DASDL UPDATE run. In particular, when the level number is not increased, it will facilitate crossing software release levels.

The level will now only be increased if certain "significant" changes occur. Such changes include adding and deleting structures, any reorganization, and replacing filler with data. When the level is increased, a new control file must be made and all tailored software (ACCESSROUTINES, RECOVERY, Data Recovery, UTILITY, Saved Reorganization Programs) must be recompiled. When the level is maintained, only those items affected need be remade. DASDL will report which items are affected. At such a time, more than one set of ACCESSROUTINES will be able to run and the user will be responsible for ensuring that the desired set is loaded under the proper name on the correct family, etc.

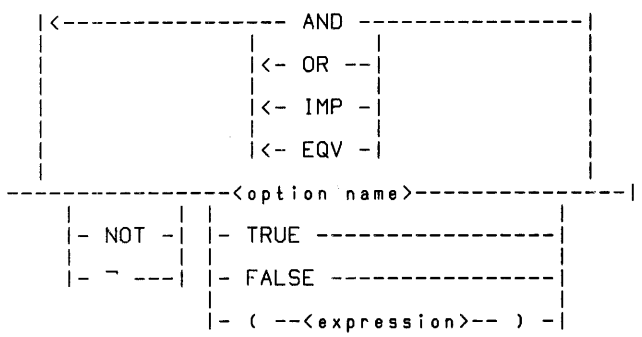
D2461 DASDL - USER DEFINED DOLLAR OPTIONS

User defined dollar options have been implemented. They are defined by their first appearance on a dollar card; by default, the value is FALSE. They are "declared" implicitly. Both standard and user options may be assigned values determined from Boolean expressions composed of options.

<option assignment>

--<option name>-- = --<expression>--|

<expression>



Note: The operators are listed in the above diagram in order of precedence.

The "= <expression>" syntax is valid only when the dollar card action is SET. User option identifiers may be up to 17 alphanumeric characters; hyphens are not permitted. Incorrect dollar card syntax will now produce syntax error messages from the compiler.

D2534 DASDL - "UPDATE" AND "INITIALIZE"

DASDL now ensures that INITIALIZE is not specified during DASDL update. This prevents inadvertent reinitialization of the data base.

DMS II - DASDL

P1028 DASDL - "DASDL" UPDATE TIMESTAMP

The process of performing DASDL UPDATE's is intended to be linear. A new set of checks has been implemented to prevent "branching". The current (II.9) implementation requires that programs, accessroutines and control files match on the basis of the data base timestamp and update level. Since the update level is simply a counter, it was possible to create two different descriptions which had the same timestamps and update levels.

An update timestamp has been added to the description file and passed along to code files and control files which will tie all of them together more tightly. There will be a new value of the timestamp for each update level, and any checks currently made on the update level will be accompanied by a check on the update timestamp to make sure that it corresponds exactly.

The propertylevel, siblevel and acrlevel values have all been increased in order to be able to detect which files have the new timestamps and which files do not.

DMSOPEN will handle the SIB format level 6 which contains a DASDL UPDATE level timestamp. This timestamp is passed to the ACCESSROUTINES and compared to a value in tables which have been compiled into the ACCESSROUTINES.

P1059 DASDL - "PARTITIONINFO" IN COMPILE ORDER LIST

The PARTITIONINFO data set is now put into the compile order list after global data. This prevents a possible syntax error when compiling the ACCESSROUTINES.

P1060 DASDL - GLOBAL DATA "CHECKSUM"

Specification of the CHECKSUM option is now effective. If UPDATE errors occur in DASDL because of this change, corrective action would be to remove the specification or to specify REORGANIZE GLOBAL DATA to add the checksum.

P1203 DASDL - GROUP KEYS AND "REORGANIZE"

It is no longer permitted to rearrange the items of a group which is a key when "REORGANIZE(ITEMS CHANGED)" is specified. Such an action would invalidate existing set entries and programs using it, even with remaps.

P1204 DASDL - "UPDATE" CHANGE OF DATA SET SUBTYPE

The compiler no longer permits the change of data set subtypes (UNORDERED, DIRECT, etc.) when doing an UPDATE compile.

P1205 DASDL - ERROR RECOVERY IMPROVEMENT

The handling of certain instances of incorrect syntax input has been improved. In particular, faults no longer occur in some cases where an identifier was expected but the check for the identifier was incorrect.

P1206 DASDL - COMPILER FAULT

An INVALID OP fault no longer occurs when designated serial number specifications are made for the audit trail.

P1207 DASDL - MACHINE "ID" IN HEADING

DASDL will now use the TIME(23) intrinsic to print the machine identification on the heading of the line printer listing. Previously, B6700 was printed unconditionally.

P1208 DASDL - DELETE VARYING ITEMS

Checking of variation options (STORED, OCCURS, SIZE) on a DASDL UPDATE compile has been corrected to permit items which use the options to be deleted from a data set record.

P1275 DASDL - IMPROPER INVALID TEXT

For required alpha items with a null value of "HIGH-VALUE", the INVALIDTEXT was generated improperly. SEG ARRAY faults would occur during record deletions. The nature of the error was not such that data corruption or record recognition problems would occur. This problem has been corrected.

P1446 DASDL - NUMERIC ITEM SIZE OF ZERO

A syntax error is now given if a numeric item is declared with a size of zero.

P1447 DASDL - DELETING MANY STRUCTURES

DASDL no longer terminates with the message "DESCRIPTION TABLE SIZE EXCEEDED" when several hundred structures are deleted.

P1472 DASDL - LABEL SOURCE INPUT CORRECTLY

Following an INCLUDE, all source was labeled with a "D" rather than a "C". This problem has been corrected.

P1473 DASDL - "NEWTAPE" FILE IN ERROR

When the LISTP and NEW options are set and a syntax error is detected, the line containing the error will no longer be placed in the NEWTAPE file twice.

P1474 DASDL - "UNIQUE" COMPILES CORRECTLY

DASDL no longer produces a spurious syntax error when a data item includes the "UNIQUE" option.

P1526 DASDL - "INCLUDE" DOLLAR OPTION

DASDL now issues an appropriate error message if the title of the included file is not enclosed in quotes. The selection mechanism has also been corrected so that any portion of a file may be included.

P1527 DASDL - DEFAULT INITIAL VALUE

DASDL will now correctly initialize Boolean, numeric and real items to the value specified in the item default statement.

P1528 DASDL - "INITIALVALUE" IN "DEFAULT" STATEMENT

When an INITIALVALUE was specified in a DEFAULT statement, an invalid initial value was assigned to some items. This problem has been corrected.

P1529 DASDL - PROPERTY BUFFER DECLARATION

The size of the data base properties buffer has been increased by one word to prevent spurious SEG ARRAY errors.

P1530 DASDL - BUILD "CONTROLITEMNODE" CORRECTLY

If a data base contained variable format records, the CONTROLITEMNODE in the data base description file was not constructed properly. This could result in a VERSIONERROR when a link item was referenced at run time. This problem has been corrected.

P1584 DASDL - SPURIOUS ERROR DURING "DASDL" UPDATE

If a data set containing a GROUP item was reorganized using the ITEMS CHANGED option, a spurious update error could be reported. This has been corrected.

P1588 DASDL - PREVENT LOOP ON DOLLAR CARD

DASDL will no longer fall into an infinite loop when it encounters a $ SET HDR or $ SET PAGE dollar option card.

P1617 DASDL - FREE PROPERTY BUFFER FOR ACCESSES

DASDL will no longer terminate with a software error at 30106500 when a data base with many accesses is compiled.

P1686 DASDL - PREVENT SPURIOUS MESSAGE

DASDL no longer displays the spurious message "<data base name> UNASSIGNED DELETED" during update. The description file which DASDL produced was correct despite the message.

P8913 DASDL - INVALID AUDIT "AREASIZE" WITH "SEGMENTS"

DASDL will no longer produce an erroneous AREASIZE when the audit trail AREASIZE was specified in "SEGMENTS".

P8914 DASDL - REORGANIZATION OF DATA SET

Reorganization information handling within the DASDL compiler has been corrected for data sets with accesses.

P8916 DASDL - EXPAND DESCRIPTION FILES

DMS II will now handle data bases whose descriptions exceed 65536 words.

P8950 DASDL - VARIABLE FORMAT REMAPS

A syntax error will now be given when an item from the fixed part of a data set record is included in a variable part of a remap. Previously, the error was given only when the properties of the item were respecified.

P9053 DASDL - "VERIFYSTORE" TEXT FOR REMAPS

The VERIFYSTORE text for remaps is now generated correctly for certain instances of HIDDEN and READONLY items. The effect of the problem would have been erroneous DATAERROR exceptions.

P9074 DASDL - OPTIMIZE STORAGE OF INITIAL AND NULL VALUES

The common storage of BLANK and ZERO initial and null values in the description file has been corrected.

P9088 DASDL - MINUS SIGN PROBLEMS

DASDL no longer carries a minus along to numbers following a negative number in in a condition. The following example now produces the proper text:

```
X LSS -1 OR X GTR 1
```

P9089 DASDL - SETS WITH NAME "DATA"

DASDL no longer permits a set or subset to have the name "DATA". Using this name would result in the file titles of the set and data set being the same. An error message is now given.

P9090 DASDL - "DIVIDE BY ZERO"

DASDL no longer does a divide by zero when computing the default areasize of a data set whose blocksize is one word.

P9091 DASDL - NON-USERCODE DATA BASES

DASDL will now put the description file and utility, recovery, datarecovery and reconstruct programs under the system directory (rather than the running usercode) when the following is specified:

```
COMPILE *DBNAME
```

Formerly, this condition was only applied to the ACCESSROUTINES.

P9095 DASDL - COMPILE RECONSTRUCT WITH "DBNODE" DEFINE

DASDL now uses a define for the location of the "DBNODE" when zipping the compile for reconstruction.

P9186 DASDL - GROUP KEYS FOR RANDOM/INDEX RANDOM

Certain keys for random accesses and index random sets were not being handled correctly. The affected cases were those keys whose last element was a group which in turn had its last element end on an odd digit boundary.

```
Examples:  G1 GROUP (M NUMBER(3); N NUMBER(2); );
           G2 GROUP (M NUMBER(3); A ALPHA(2); N NUMBER(3); );
```

In such cases, selection expressions using the items of the group (FIND AT M = value AND N = value) could fail to find existing records with the specified keys.

Random and index random, only, are affected because of their folding of the entire key. The correction of the problem will change the blocks in which keys are to be found. Therefore, for existing data bases with such structures, the following corrective procedure should be followed:

1. DASDL update compilation, using new compiler.
2. Generation and execution of reorganization program, regenerating index random sets from the data sets they span.
3. Recompilation of ACCESSROUTINES.

P9252 DASDL - "NOT FOUND" EXCEPTIONS IF ENTIRE KEY GIVEN

Improper NOT FOUND exceptions are no longer generated for certain FIND operations. The problem occurred when all elements of multiple-item key were specified and the key ended with signed or descending items.

P9262 DASDL - AUTOMATIC SUBSETS OF SETS

In certain cases, the key structure of a set which had an automatic subset specified against it was being set up incorrectly. The problem has been corrected.

If an existing data base has an affected set, UPDATE using a new DASDL compiler (with this change) will be unsuccessful. In such cases, the set must be deleted from the data base, added back and regenerated via REORGANIZATION.

P9263 DASDL - PRINTING FILE TITLES

When the STORE or FILE (for III.0 only) dollar option is set, a question mark no longer appears at the end of titles of files on system resource pack.

D2219 DMALGOL - NODE SYNTAX

The DMALGOL node declaration "NODE X*" has been eliminated. This construct was required because of rather unfortunate semantics of "N[I]", where N is a node. It was necessary to use "X[N[I]]" to retrieve the contents of N[I]. Now the semantics of N[I] does in fact produce the contents of N[I]. To compensate for this change, it is necessary, in most cases, to merely remove the "X*" declaration and change references like "X[N[I]]" to "N[I]". Where constant subscript values were used, new properties have been defined for use instead.

D2276 DMALGOL - PRINT "D1" STACKSIZE

When compiling ACCESSROUTINES, the D1 stack size is printed along with the D2 and D3 stack sizes at the end of each structure environment. Note that while the D2 and D3 values give the stack sizes for each environment, the new D1 value is the accumulated D1 stack size for all structures compiled so far.

DMS II - DMALGOL

P1606 DMALGOL - DATA POOLS IN "I/O" LISTS

It was possible for the presence of long strings in I/O lists to cause system dumps at FORGETCHECK at run time. This has been corrected.

DMS II - DMCONTROL

D2435 DMCTL - UNAUDITED DATA BASE CONTROL FILE RECOVERY

The "RECOVER" functions of DMCONTROL are now permitted for unaudited data bases. They are intended for use only under very limited circumstances. In all cases, the user should have extra knowledge about the state of the data base. In general, when a control file is lost, the only safe action is reloading and reprocessing (for unaudited data bases). There are some cases where making a new control file is all that is ncessary (e.g., the control file is on a pack separate from the other files and that pack went bad). It is for such cases that these functions are permitted.

D2436 DMCTL - HALT/LOAD BIT OVERRIDE

A way of resetting the Halt/Load bit (also called the data base in-use bit) for unaudited data bases has been provided. It is accomplished by doing the following:

```
RUN SYSTEM/DMCONTROL ("OVERRIDE HL");
FILE CF=<control file name>;
```

The result will be a control file with the bit reset to zero.

In general, the Halt/Load bit being on raises questions about the integrity of the data base; therefore, this function is intended only for use in very limited circumstances when the user has more knowledge about the state of the data base than does the control file. It is not a general alternative to normal recovery via reloading and reprocessing. Using this function requires the utmost care; this function not be used when there is even the slightest doubt about the state of the data base.

Error messages will be produced if the data base is audited or the bit is not on initially. DMCONTROL will require exclusive use of the file in order to perform this function.

D2458 DMCTL - VERSION OVERRIDES RECORDED

The date of the occurrence of a version mismatch override will now be recorded in the control file and be made to appear in data base software program dumps.

DMS II - DMCONTROL

P1021 DMCTL - BETTER DISPLAYS FOR CONTROL FILE CHECKS

The control file module now emits more informative displays for control file/DB software mismatches.

P1022 DMCTL - INVALID CONTROL FILE CREATION FOR PARTITIONS

The control file is no longer built incorrectly for partitioned data bases. Formerly, DMCONTROL may have dropped some partition entries from the control file.

P1276 DMCTL - INVALID VERSION TIMESTAMP MISMATCH

The control file module no longer emits an invalid "WRONG VERSION OF FILE ..." when two system software programs are running (e.g., UTILITY and ACCESSROUTINES, DATARECOVERY and ACCESSROUTINES, etc.). To effectively use this change, SYSTEM/DMCONTROL must be recompiled first and a new DATABASE/DMCONTROL (symbol) must be made including this change. Only UTILITY and DATARECOVERY must be recompiled after this (including the new DATABASE/DMCONTROL).

P1347 DMCTL - SEPARATE INCLUDE OF "CF, CFOLD"

A program can now include the declarations and defines for CF and CFOLD at separate places in the symbolic.

P1349 DMCTL - CHECK FOR VALID STRUCTURE NUMBER

A new case has been added to the CF module so that it is possible to tell if a structure number is valid. UTILITY uses this check when setting up its directory of files to be reconstructed so that it does not include deleted files.

P1636 DMCTL - REBUILD FROM "II.8" DUMP TAPES

DMCONTROL will now initialize creation timestamps in the control file to zero under the "INITIALIZE 29" command. This will allow post-II.8 rebuild software to work with II.8 dump tapes.

P9082 DMCTL - OPTIONAL FILE WHEN INITIALIZING CONTROL

When updating the description file for the first time from II.8 to II.9 or II.8 to III.0, DMCONTROL will initialize the control file. During the initializing process, DMCONTROL will also attempt to initialize DMTIMESTAMP in all existing data base files. This permits the user to respond as follows when a data base file is not present (excluding restart data set):

AX:CONTINUE

If the user utilizes this option, a time stamp mismatch will occur when the data base software opens this file(s) at run time. (This mismatch may be overridden at run time.)

P9092 DMCTL - OPTIONAL PROGRAMDUMP FOR CONTROL FILE ERRORS

The control file module will now ask the user whether or not a programdump is desired for certain control file errors.

P9301 DMCTL - CONTROL FILE COMPATIBILITY ACROSS RELEASES

The control file module now handles the control file format differences across releases. The II.9 module now accepts III.0 control files for all data base software (provided update level, data base timestamp, etc., are compatible). The III.0 module accepts II.9 control files for the following cases:

1. Description (DASDL) update from II.9 to III.0.
2. Control file (DMCONTROL) update from II.9 to III.0.
3. Control file (DMCONTROL) recovery from II.9 to III.0.
4. Control file handling from utility dump tapes (UTILITY).

DMS II - HELPINQUIRY

D2459 HELPINQ - NEW "HELP" FILE

An entirely new DMINQ/HELP file has been created. All the syntax diagrams and semantic explanations are new, using upper and lower case characters (lower case will appear as upper case on terminals with no lower case capability).

Also, the information has been updated to include III.0 features.

DMS II - HELPINQUIRY

P9087 HELPINQ - CORRECT SYNTAX DIAGRAM

A minor error has been corrected in the syntax diagram produced by the HELP verb for the SORT option of SELECT/DISPLAY. SORT and REPEAT may appear in any order, rather than SORT first, as implied by the incorrect syntax diagram.

D2251 INQ - "QUIT"

Since it may take several seconds to terminate an INQUIRY session after typing QUIT, the INQUIRY program now responds with "TERMINATING INQUIRY" just prior to closing the data base.

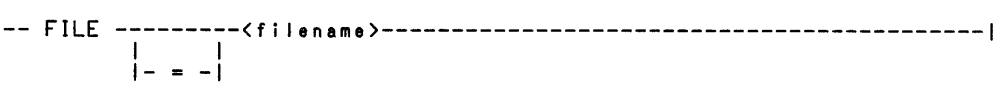
D2408 INQ - PRINTER FILE ATTRIBUTES

Primarily as a result of new report features in INQUIRY, it has become desirable to be able to control file attributes for the printer file used by INQUIRY. Such attributes include, but are not limited to, formmessage, kind, etc.

The new feature is an extension to the INQUIRY statement "PRINTER". A new option, "FILE", is used to change the intname attribute of the INQUIRY printer file. This will associate with the printer file any file attributes specified for that file name at the time the INQUIRY program is initiated.

Note that the user must anticipate any special requirements for the printer file. This need not be a serious drawback, since it seems likely the user can anticipate the creation of a report requiring, for example, special forms. It should be clear that this feature is not required to produce normal printer files, but only when the user wants to do something "unusual".

The syntax for the new printer option is:



where <filename> is an identifier.

If the printer file is already open, it is closed. Then, the title of the printer file is set to the given <filename>; finally, the intname is set to the <filename>. If no file-equation WFL statement had been supplied for this file name, the file reverts to the default printer file with the given <filename> as its title. Note that the original intname of the printer file is "PRINTER". The file is not actually opened until some INQUIRY statement requires it, so this process may be repeated as necessary. The use of the "PRINTER" INQUIRY statement with no options displays the current setting of all options, including the new file option.

D2409 INQ - MULTIPLE-STATEMENT DEFINES

I. General

The original DMINQ define mechanism permits the user to "define" anything that can be entered with a single input. This can range from a simple identifier synonym to an entire DMINQ statement. Since statements are always terminated by "END-OF-LINE", there can never be more than a single statement per input; thus a define cannot contain more than one entire statement.

An extension to this define mechanism has been implemented which permits a define to contain several statements. A side-effect of this extension is the ability to enter more than a single statement with one input for immediate execution, not just as the text of a define.

The basic idea is to use a special delimiter, the semicolon (";"), as a separator between two statements entered with one input. However, there are certain restrictions and requirements which apply when declaring and invoking multiple-statement defines:

1. A multiple-statement define must contain complete statements. This does not exclude the invocation of other defines; however, any statement "started" in a multiple-statement define cannot be "completed" in another multiple-statement define.

2. A multiple-statement define may be invoked only where a statement may occur; that is, first in an input or multiple-statement define text, or after a semicolon.

3. The invocation of a multiple-statement define must be last in an input or multiple-statement define text, or must be followed by a semicolon.

4. A multiple-statement define may not contain a define, EDIT, or REPEAT-with-EDIT statement.

It may be helpful to think of a multiple-statement define as a "subroutine", containing complete statements; an invocation of a multiple-statement define is itself a statement, just like a subroutine "call". However, like any define, the multiple-statement define is not parsed and verified until its invocation, so violation of these rules, or other syntax errors, may not be detected until later.

Changes have also been made in the syntax and/or semantics of related verbs to facilitate modification of multiple-statement inputs and defines (see below).

II. Syntax

A <basic-statement> is any INQUIRY statement except a define, EDIT, or REPEAT-with-EDIT statement.

A <msd-invoke> is an invocation of a multiple-statement define (the define identifier together with parameters, if any).

A <logical-statement> is any <basic-statement>, or a <msd-invoke>.

The <text> of a multiple-statement define is two or more <logical-statement>s, separated by semicolons (and optionally followed by a semicolon), or a single <logical-statement> followed by semicolon. The <text> of an ordinary define never contains a semicolon.

Since the <text> of a multiple-statement define may not contain a DEFINE, EDIT, or REPEAT-with-EDIT statement, these statements occur only in "normal input", and are terminated only by the "END-OF-LINE".

If an error occurs during the execution of any statement in a multiple-statement define, at any level of invocation, all remaining statements in invoked multiple-statement defines are ignored and the system waits for the next user input. It is assumed that the statements are designed for the "normal" situation, and that an error is abnormal.

III. Other changes

In order to facilitate the modification of multiple-statement inputs in general, and of multiple-statement defines in particular, changes have been made in the syntax of EDIT (and REPEAT-with-EDIT), and in the semantics of SHOW and RECALL.

Whenever the "SAVETEXT" buffer contains a multiple-statement define or a multiple-statement input, "SHOW" will display each statement on a separate line, preceded by a line number and a colon (:). Similarly, since "RECALL" always performs a SHOW after placing the requested text into the savetext buffer, a RECALL of a multiple-statement define will also break the text up into separate lines.

An extension to EDIT makes it possible to EDIT only the text of a specific line. It is merely necessary to enter the line number prior to the first delimiter. The line number must correspond to one of the lines displayed with the SHOW, or a syntax error will result. The line numbers are never saved, but merely indicate the position of the line within the entire text. If any change inserts new statements, or deletes current statements, the line numbers of succeeding lines will, or course, change. Once again, an EDIT operation will perform a "SHOW", so the new text and new line numbers will be displayed.

Note that EDIT does not REPEAT, or re-process, the modified text. Thus a sequence of modifications may be lost if REPEAT is not entered prior to other input. The REPEAT-WITH-EDIT statement may be used to cause the new text to be processed immediately.

IV. Examples

```
DEFINE D1(X,Y) = CLEAR X; VIRTUAL X = N OF D + Y; SHOW X
DEFINE D2(X,Y) = VIRTUAL X = N OF D + Y
DEFINE D3(X,Y) = VIRTUAL X = N OF D + Y;
```

The defines D1 and D3 are multiple-statement defines (since they contain at least one semicolon); as such, they may be invoked only as complete statements. D2, even though it appears to be a complete statement, is not a multiple-statement define because it contains no semicolon. It may be invoked as part of a larger statement, such as

```
D2(V2, 2) - 10
```

Which is equivalent to

```
VIRTUAL V2 = N OF D + 2 - 10
```

If virtuals V1, V2 and V3 are declared by the statements

```
D1(V1,1)
D2(V2,2) -10
D3(V3,3)
```

Then RECALL V1 will display

```
VIRTUAL V1 = N OF D + 1
```

And similarly, RECALL V3 will display

```
VIRTUAL V3 = N OF D + 3
```

However, RECALL V2 will display

```
D2(V2,2) -10
```

RECALL DEFINE D1 will display the following:

```
01:DEFINE D1(X,Y) = CLEAR X;
02:VIRTUAL X = N OF D + Y;
03:SHOW X
#
```

EDIT /D/Z will produce:

```
01:ZEFINE D1(X,Y) = CLEAR X;
02:VIRTUAL X = N OF D + Y;
03:SHOW X
#
```

Note that the first "D" of the entire text was changed (normal EDIT). However, EDIT 2/D/Z will change the first "D" of line number 2:

```
01:DEFINE D1(X,Y) = CLEAR X;
02:VIRTUAL X = N of Z + Y;
03:SHOW X
#
```

D2410 INQ - REPORT FACILITY FOR "DMSII INQUIRY"

This document describes an interactive REPORT option for the DMSII INQUIRY system.

The REPORT option is intended to be used to produce simple reports from information held in DMSII data bases. The COBOL report facility or the "REPORTER SYSTEM" (document #1079555) can be used to generate complex reports or reports for non-DMSII data.

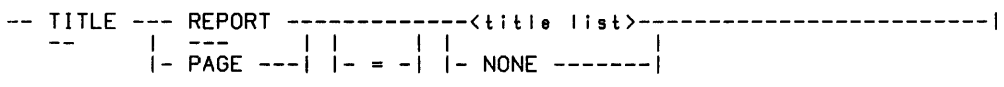
Report definitions can be saved through sessions by utilizing the multi-statement define mechanism.

REPORT OPTION

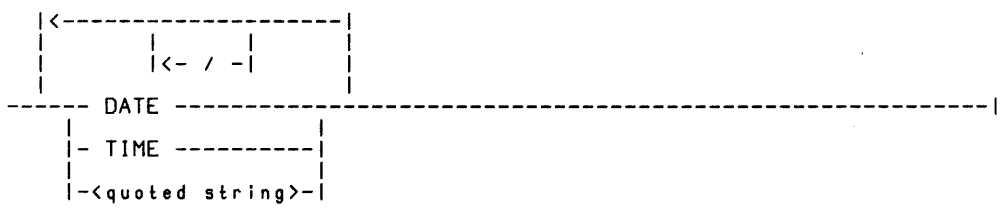
The REPORT option consists of three new statements TITLE, REPORT and SUMMARY plus additions to three existing statements RECALL, CLEAR and GENERATE.

REPORT STATEMENT DETAILS

REPORT/PAGE TITLES



<title list>



1. REPORT will produce a title page for the report. The title will be centered on the page. A report title page is optional.

2. PAGE will produce a title on the top of every page of the report. A page title is optional.

3. DATE will print the date as part of the title in the form:

   MONTH/DAY/YEAR Example: 9/19/76

4. TIME will print the current time as part of the title in the form:

   HOUR:MINUTE Example: 9:45

5. The length of the quoted string cannot exceed line width.

6. A slash (/) may be used to produce multiple line titles.

7. The option NONE will eliminate a previous title.

   Example:

      TITLE PAGE "A PAGE TITLE" / DATE / TIME

   will produce on each page

```
                    A PAGE TITLE
                      9/19/76
                       8:30
```

REPORT VERB

The report verb controls which items are listed on the report. In addition it provides control over the format of the report.

```
-- REPORT --<report list>---------------------------------------------|
   ---
```

<report list>

```
---------------------<id list>----------------------------------------|
  |                 |
  |-<control list>-|
```

<control list>

```
  |<-------------- , --------------|
  |                                |
---------------<id>------------------ : ------------------------------|
     |        |        |          |
     |- PAGE -|        |- TRUNCATE -|
                          ----
```

<id list>

```
  |<------------------------ , -----------------------|
  |                                                   |
----<id>--------------------------------------------------------------|
         |              | |             | |            |
         |-<colstart>-| |-<colwidth>-| |- TRUNCATE -|
                                           ----
```

<colstart>

```
---- @ ----<integer>--------------------------------------------------|
  |    |
  |- AT -|
```

<colwidth>

```
---- # -------<integer>-----------------------------------------------|
  |        |
  |- WIDTH -|
  |        |
  |- FOR ---|
```

<id>

```
---------------------<item>-------------------------------------------|
  |                 |
  |-<colname>-- = -|
```

GENERAL

Item may be any of the following:

1. Any data base item name (but not a group name). If the item has an OCCURS, the subscript may be an arithmetic expression.

2. Virtual item.

3. An arithmetic function.

4. An arithmetic expression.

The <colname> option allows the user to specify the name that will be printed on the report. If <colname> is not specified, the item name will be used as the column name. <colname> (which cannot exceed 17 characters) can be any alpha-numeric literal which begins with an alphabetic character.

No attempt is made to "qualify" the items, expressions, virtuals, etc., which appear in the <report list>. Reference to items in "NOT-FOUND" data sets will be displayed as undefined (two hyphens). Referencing items in data sets which are not selected will result in cancelling the report with a "NOT SELECTED" error.

A. <control list>

Items appearing in the control list are considered to be control items. A control break occurs when the value of any control item changes between two adjacent records. (See SUMMARY statement.) To provide proper report generation control, the selected records should be SORTed in order of the control items, if the selection is not already in that order. The SET statement should be used to specify the sort criteria to avoid redundant calls on the system sort intrinsic.

Example:

(USER INPUT)

```
REPORT CLASS:NAME,YARDS,TD=TOUCHDOWNS
SELECT VIA CLASSET AT CLASS="PASSING" OR CLASS="RUSHING"
GENERATE REPORT VIA HALL-OF-FAME
```

(SAMPLE PAGE RESULT)

```
------------------------------------------------------------
                                                  PAGE 1

CLASS:PASSING

     NAME                YARDS         TD
     ----                -----         --
     UNITAS,JOHNNY       40239        290
     TARKENTON,FRAN      38840        291
     TITTLE,Y.A.         33070        242
     -
     -

CLASS:RUSHING

     NAME                YARDS         TD
     ----                -----         --
     BROWN,JIM           12312        106
     PERRY,JOE            9723         71
     TAYLOR,JIM           8597         83
     -
     -

------------------------------------------------------------
```

Note that items in the <control list> are printed in the form:

NAME:VALUE

and the column heading is repeated.

The TRUNCATE option is ignored for non-alpha variables and for alpha variables where the number of characters in the variable will fit on the remainder of the line.

If TRUNCATE is specifed for an alpha variable which will not fit on a line, the left most charaters (sufficient to fill the line) will be printed and the remaining characters will be lost.

IF TRUNCATE is not specified, the entire alpha variable will be printed taking multiple lines; if necessary, however trailing blanks will be suppressed.

More than one control item may be specified. In addition, if the option PAGE precedes a control item, a page eject will also occur when the control item changes value.

The PAGE option indicates that a new page should be started if a control break occurs.

Example :

```
REPORT PAGE DEPT,SECT:NAME,ROOM
```

RESULT:

```
-------------------------------------------------------
                                              PAGE 3
DEPT:6700
SECT:6710

     NAME          ROOM
     ----          ----
     DOE,JOHN      421
     -

     -
SECT:6720
     NAME          ROOM
     ----          ----
     SMITH,JOHN    527
     -
     -

-------------------------------------------------------
                                              PAGE 4
DEPT:6800
SECT:6810

     NAME          ROOM
     ----          ----
     DOE,JOHN      435
     -
     -

SECT:6810

     NAME          ROOM
     ----          ----
     DOKES,MARY    645
     -
     -

-------------------------------------------------------
```

B. <id list>

1. Those items specified in the <id list> will be printed in a "columnized" format.
2. The <colstart> option allows the user to specify the starting position of a column on the report page.
3. If <colstart> is not specified, it is assumed that the column will start one character beyond where the prior column finished.
4. A "COLUMN OVERLAP" error will be displayed if <integer> specifies a position less than the position beyond the end of the prior column.
5. The <colwidth> option allows the user to specify the number of character positions in which a value is to be formatted.
6. If <colwidth> is not specified, a column width is assumed to be the largest of either the number of characters in the column name or the number of characters it takes to present the data value without loss of significance. This may result in a "REPORT TOO BIG FOR LINE" error.
7. If the <colwidth> specified is not large enough to hold the column name, the <colwidth> is ignored and a column width large enough to hold the column name is assumed.
8. The <colwidth> may be specified much larger then necessary to space the items evenly across the report page.
9. If the <colwidth> specified is not large enough to hold the data to be formatted, the action depends on the data type.

   NUMERIC: The value will be rounded to fit within the <colwidth> specified. If this cannot

be done without loss of leading significant digits, <colwidth> will be filled with * characters.

BOOLEAN: The left most portion of the TRUE or FALSE will be placed in <colwidth>.

ALPHA : See the TRUNCATE option

C. TRUNCATE option:

The TRUNCATE option is ignored unless a <colwidth> is specified.

If the TRUNCATE option is not specified, the entire alpha variable will be printed even if it takes multiple report lines to do so.

EXAMPLE:

```
REPORT BIGALPH:A-NUM,CMMT#19,ANOTHER-NUM,CMMT-2#18
```

RESULT:

```
BIGALPH: A LONG ALPHA VARIABLE USED AS A CONTROL ITEM AND WHEN TRUNCATE
         IS NOT SPECIFED, WILL RESULT IN MULTIPLE LINES OF PRINT, IF
         NECESSARY, TO PRINT THE ENTIRE ALPHA VALUE

      A-NUM CMMT                 ANOTHER-NUM CMMT-2
  --------- -------------------- ----------- ----------------
      23.66 THIS IS AN ALPHA                1 MULTIPLE LONG
            VALUE WHICH WILL                  ALPHA IS ALLOWED
            TAKE MORE THAN ONE                IN THE SAME REPORT
            LINE TO PRINT                     LIST
     104.73 WE WILL ALWAYS TRY              2 A SHORT ONE
            WHEN POSSIBLE TO
            SPLIT AT A BLANK
     108.98 A SHORT ONE                     3 NOTE EITHER LONG
                                              ALPHA MAY CAUSE
                                              MULTIPLE LINES
     108.97 TRAILING BLANKS ARE             4 NEXT IS NULL
            SUPPRESSED
     108.98 ANOTHER SHORT ONE               5 --
```

If the TRUNCATE option is specified, and the column width is smaller than the number of characters in the alpha variable, only the left most <colwidth> characters are placed in the column.

EXAMPLE:

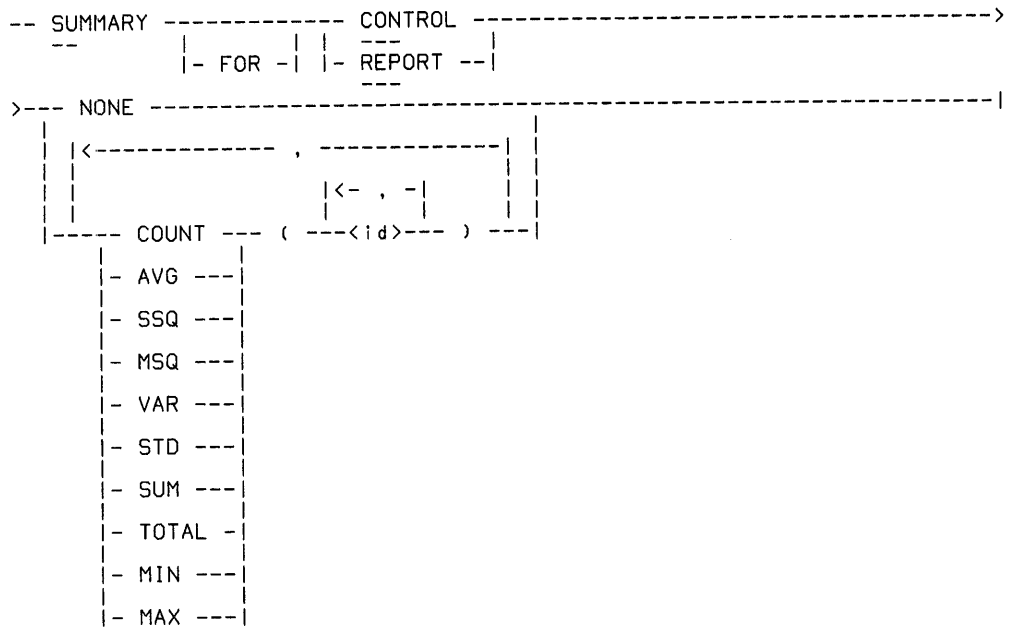
```
REPORT BIGALPH TRUNCATE :A-NUM,CMMT #19 TRUNCATE,ANOTHER-NUM
```

RESULT:

```
BIGALPH: A LONG ALPHA VARIABLE USED AS A CONTROL ITEM AND WHEN TRUN

      A-NUM CMMT                 ANOTHER-NUM
  --------- -------------------- -----------
      23.66 THIS IS AN ALPHA VA            1
     104.73 WE WILL ALWAYS TRY             2
     108.98 A SHORT ONE                    3
     108.97 TRAILING BLANKS ARE            4
     108.98 ANOTHER SHORT ONE              5
```

SUMMARY VERB

The SUMMARY verb provides statistical information about items in the report list.



1. <id> must reference an <id> in the <report list>. If the <colname> is used in the <report list> for an item, <id> must be the <colname>.
2. TOTAL and SUM are synonymous.
3. The REPORT statement must be entered before entering a SUMMARY statement.
4. The option NONE will eliminate a previous SUMMARY statement
5. For SUMMARY CONTROL, <id> must be in <id list>
6. For SUMMARY REPORT, <id> can be in either <control list> or in <id list>.
7. Both SUMMARY REPORT and SUMMARY CONTROL can be used within a single report.

A. Summary for Control

A control break occurs each time a control item in the report list changes value. At this time any specified summary control information is processed and displayed.

Example:

```
TITLE PAGE "REPORT"
REPORT D1:E1,E2
SUMMARY FOR CONTROL AVG (E1,E2), SUM (E1)
SET D TO SELECT D1>0,SORT ON D1
SET E TO SELECT E1>0,SORT ON E1
ATTACH E
GENERATE REPORT VIA D
```

RESULT:

```
                        REPORT                  PAGE 45
    D1 : 23

         E1        E2
         --        --
          1         2
          3         4
         ==        ==
    AVG   2         3
    SUM   4
```

B. Summary for Report

The report summary is printed as an extra page after the report is generated.

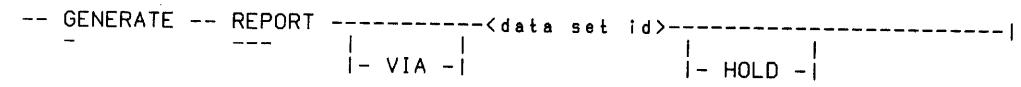
Example:

```
SELECT VIA D
REPORT D1,D2,D3
SUMMARY FOR REPORT AVG (D1), SUM (D1,D2)
GENERATE REPORT D
```

RESULT:

```
------------------------------------------------
REPORT SUMMARY

     AVG D1 = 148.92
     SUM D1 = 72, D2 = 57



------------------------------------------------
```

GENERATE REPORT



The GENERATE REPORT statement will cause the generation of the report.

The selection criteria for <data set id> must have been previously established by SELECT, SET and/or ATTACH verbs.

If multiple copies of the report are desired, this statement may be repeated as many times as desired.

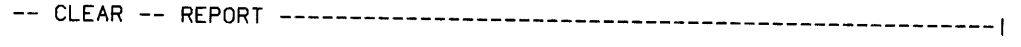
Note: If batch programs are concurrently updating the data base, each report may be different.

The report generation process can be cancelled by entering

```
?AX
```

If the HOLD option is used, the report generation process is started, but the first output line is held until the user enters a null input. This is to allow a hardcopy terminal to be positioned at the top of a page; e.g., TC500, in order that the physical page control is synchronized with the report generation process.

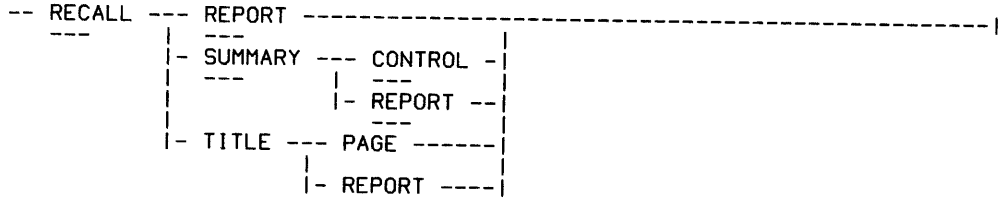
CLEAR REPORT



The CLEAR REPORT statement returns all internal storage used to "remember" the report statements and generate the report. It is desirable to enter this statement for efficiency reasons, as well as to avoid possible conflict between a previous report definition and a new report definition.

RECALL

The following RECALL options can be used to recall, edit and correct previously-entered REPORT option statements.



In some cases, changes to the report list may require changes to the SUMMARY statement.

FORMS MEDIA CONTROL

A. Physical Attributes

1. If option TERMINAL is set, the report will be generated to the user's terminal. The page size and line width will be assumed to be that of the terminal attributes width and page (see verb TERMINAL in the DMSII INQUIRY Manual). For hardcopy terminals or when connection has been established by dialin, the terminal attributes WIDTH and PAGE should be checked/set prior to generating the report.

2. If option PRINTER is set, the report will be generated to a site line printer. The page size and width used in the report generation process will be assumed to be that of the printer width and page attributes (see verb PRINTER in the DMSII INQUIRY Manual).

The terminal or printer attributes must be established prior to generating the report.

B. Page Control

The REPORT option, when detecting a new page is necessary, either due to a page eject or the page is full will:

1. If option PRINTER is set, a skip to channel 1 will be executed. The carriage control tape on the printer must be coded accordingly.

2. If option TERMINAL is set,

   A. Sufficient blank lines are written to ensure a full page.

   B. If SCREEN is FALSE, four more blank lines will be printed.

   C. If SCREEN is TRUE, the system will stop until either a null input or blank line is entered by the user.

D2581 INQ - RIGHT-JUSTIFY NUMERICS IN "DISPLAY" VERB

Values of items for the HEADING format of the DISPLAY verb are now printed right-justified in the column instead of left-justified. As a result for most numeric items, value in columns will be aligned by decimal point. Also, the heading format of DISPLAY is now consistent with the format of the INQUIRY REPORT option.

D2587 INQ - ERROR MESSAGES

The error message which explains an error is now displayed unconditionally. Previously, it was displayed only after the user entered HELP. HELP continues to work as before.

DMS II - INQUIRY

P1023 INQ - CORRECT CHECK FOR NAME CONFLICT

DMSII INQUIRY will now allow define, virtual and generate identifier names even though the first six characters are the same as an INQUIRY verb name; e.g., a define SELECTXXX is now permissible.

P1061 INQ - LARGE SELECTION CONDITIONS

A selection condition containing many terms could produce an INVALID INDEX fault. This problem has been corrected.

P1062 INQ - IMPROVED SEARCH ALGORITHMS

The search algorithms used in INQUIRY to find records which satisfy a given selection condition have been improved. There are three main areas of improvement:

1. When there are several sets which could be used to access records, the system now does a much better job in picking a set which will minimize search time.

2. If a set is selected which has a key composed of several concatenated items, a much-improved binary search results if the major key items are included in the selection condition, with "appropriate" relational operators.

3. If a set is explicitly mentioned by the user (e.g., SELECT S AT ... or DISPLAY .... VIA S AT ...), a better search via that set will be performed. Previously, such a situation usually resulted in testing every record in the data set.

P1524 INQ - "SEG ARRAY" FAULT ON INPUT

A SEG ARRAY fault would occur when an excessively long line was entered after the system had filled a screen and was waiting for a "NEXT" input. This has been corrected.

P1525 INQ - QUALIFICATION RESOLUTION

When the user was asked to resolve insufficient qualification by choosing among alternatives (the "WHICH ONE?" message), a meaningless error message could result. This has been corrected.

P1536 INQ - "INQUIRY" USES SUBSETS IMPROPERLY

It was possible for INQUIRY to pick a subset for record selection instead of a full spanning set (when no set or data set was specified). This could find fewer records than actually satisfied the given condition. This problem did not exist prior to Release II.9.2 of the INQUIRY system.

P1615 INQ - LOSS OF LAST COLUMN

To prevent double spacing due to terminal wraparound, INQUIRY will write one character less on a line than the specified WIDTH; an error, however, caused data to be formatted into the last character position of a line which was lost when displayed. This has been corrected.

P1618 INQ - "GENERATE" IN A PARAMETRIC DEFINE

It was possible to get an "UNKNOWN IDENTIFIER" error when invoking a parametric define containing a GENERATE statement. This error referred to the internal form of the formal parameter. It has been corrected.

P1619 INQ - CORRECT REPORT HEADING FEATURE

A problem has been corrected which could cause the REPORT option of DMSII INQUIRY to print the page heading only on the first page of a report.

P1620 INQ - REPORT PAGE EJECT

INQ now causes a page eject on the printer after a report is complete, thus preventing the start of one report on the same page as the last page of a previous report.

P1623 INQ - RANGE TEST

A selection expression of the following form; i.e., a range test, could produce a "SOFTWARE ERROR" at run time:

K > <value> AND K < <value>

This has been corrected.

P1687 INQ - ERROR ON USER SUPPLIED SUBSCRIPT

A problem has been corrected which caused a SUBSCRIPT OUT OF RANGE error when displaying a subscripted item were the subscript was specified by the user.

P1688 INQ - DOUBLE SPACE ON "CRT" TERMINALS

A problem has been corrected whereby, under certain conditions, displaying a line in format HEADING could cause a CRT terminal with wraparound capability to double space.

P1689 INQ - "SEG ARRAY" ON SPECIAL CHARACTER

If, in certain contexts, a "special character" (non-alpha numeric) were entered where an identifier was expected, a SEG ARRAY fault would occur. This has been corrected.

P1690 INQ - "RESTORE" LARGE TEXT

It was possible for RESTORE to fail with a SEG ARRAY fault if the text being restored occupied more than 256 words. This has been corrected.

P1691 INQ - NESTED DEFINES

Nested defines were not always expanded properly. This has been corrected.

P1692 INQ - CORRECT ABBREVIATIONS, REPORT CONTROL ITEMS

The scanning of some keywords has been changed to allow keyword abbreviations.

A problem has been corrected which, under some conditions, caused a failure of the printing of the name and value of a control item.

P1693 INQ - SOFTWARE ERROR

A software error (at 64514000), which sometimes resulted from a condition of the form A=<value> AND B=<value>, where A and B were both major key items for some set, has been corrected.

P9015 INQ - "SEG ARRAY" ERROR

A SEG ARRAY error no longer occurs if an excessively long logical input (several physical inputs connected by the "%" continuation symbol) is entered.

P9016 INQ - "INVALID INDEX"

An INVALID INDEX no longer occurs if a single "%" is input to INQUIRY.

P9075 INQ - "INVALID INDEX" AT INITIALIZATION

An INVALID INDEX fault could occur at initialization of INQUIRY if structures with structure numbers greater than 127 were invoked. This has been corrected.

P9085 INQ - OPTION "QUOTES"

The value of the QUOTES option was displayed incorrectly. It printed TRUE when false, and FALSE when true. This has been corrected.

P9086 INQ - CORRECT SYNTAX ERROR

An incorrect syntax error was produced in some cases of a function being used as part of a condition of another function. This has been corrected.

P9179 INQ - DISPLAY OF EXCEPTION VALUE

Whenever INQUIRY gets an unexpected exception from the ACCESSROUTINES, the exception value is displayed and partially analyzed. Incorrect and/or incomplete analyses of that value have been corrected.

P9259 INQ - "SAVE/RESTORE" LARGE TEXTS

Any define texts occupying more then 128 words could not be SAVE'd properly. This limit is unlikely to be exceeded on II.8 or II.9, but may be exceeded with multiple statement defines on III.0.

DMS II - LOADDUMP

P1596 LOADDUMP - FILLER DROPPED FROM "COBOL" FILE DESCRIPTION

LOADDUMP no longer drops items from the generated file description when there are filler items in the source file description.

P1694 LOADDUMP - SYNTAX ERROR WHEN MAPPING BIT FIELDS

In extracting a field descriptor from a data set, a problem has been corrected in which Boolean fields would receive syntax errors upon MOVE CORRESPONDING from the field descriptor to the data set. Formerly, a bit field would be mapped into a COBOL group with the same identifier as the bit field. This would cause an implicit "MOVE" by MOVE CORRESPONDING and the resultant syntax error. A unique identifier is now defined in the field descriptor and MOVE code is generated explicitly.

P1695 LOADDUMP - INVALID SYNTAX ERROR

LOADDUMP no longer generates an invalid "VARIABLE FORMAT DATA SETS ARE DISALLOWED" when the data set specified is of fixed format in a logical data base.

DMS II - PRINTAUDIT

P1351 PRINTAUDIT - TIME OFF BY ONE SECOND

Because the algorithm was rounding instead of truncating, the printed timestamp of the audit block was sometimes off by one second. (The hex value was printed correctly.) This has been corrected.

DMS II - PROPERTIES

P1182 PROPERTIES - "DBS D3 NOMEM"

DMSOPEN will now hang with an RSVP if insufficient memory is available for the D3 stack. If this RSVP is DS'ed, DMSOPEN will return ERROR 42.

P8943 PROPERTIES - CORRECT DISCONTINUITY CHECKING

The handling of discontinuities in the audit and control file produced by OPEN INITIALIZE and REORGANIZATION has been corrected.

DMS II - PARTITION CONTROL

P1024 PTNCTL - CHANGE INCLUDE RANGE ON "DATABASE/PROPERTIES"

The include range on DATABASE/PROPERTIES has been altered to accommodate additions to PROPERTIES.

P1350 PTNCTL - WRONG FAMILY FOR GLOBAL DATA

PARTITION CONTROL ignored the family specified in DASDL for the global data file, which caused a NO FILE if that file were not on the user's family. This has been corrected.

P1209 RECOVERY - TERMINATION CONDITIONS, "ROLLBACK, REBUILD"

One of the termination conditions for ROLLBACK and REBUILD has the form:

ROLLBACK TO BOJ OF <job#>/<mix#> ON <date-time>

The <date-time> has been ignored by both ROLLBACK and REBUILD. If the <job#>/<mix#> is not found (e.g., the wrong numbers are specified), the processes will not stop until much later. This is particularly serious for ROLLBACK, which could back out (irreversibly) many desired changes. Now the <date-time> is used as an additional termination condition; the process will stop when either condition is encountered.

P1352 RECOVERY - ESSENTIAL, INESSENTIAL AUDIT RECORDS

It is now possible to add new audit records in future releases without increasing the audit level. This is accomplished by recognizing whether the audit record is essential or inessential for recovery purposes. Unrecognized inessential audit records are ignored. Unrecognized essential audit records cause an error termination of recovery. It will usually be possible for RECOVERY of one release to use audit files from a future release. This can be useful immediately after backing off from a new release after having run on it for a while. Previously, the audit level was increased on the new release, and RECOVERY from any prior release would reject all audit tapes from the new release. Now, it will normally not be necessary to increase the audit level on a new release.

P1353 RECOVERY - MISCELLANEOUS REBUILD ERROR

If the audit records for the two control points which occurred immediately prior to a recovery point record were contained entirely within one audit block, when attempting to rebuild through this section of the audit, RECOVERY would terminate abnormally with "UNIDENTIFIED MISCELLANEOUS ERROR". The error would occur on the first audit record in the audit block after the block containing the audit records for the two control points. This has been corrected.

P1354 RECOVERY - REBUILD AFTER DELETING STRUCTURE

If a utility dump were taken prior to an update which deleted a structure, a rebuild across that discontinuity could fail (INVALID INDEX, etc.). This has been corrected.

P1393 RECOVERY - ORDERED DATA SET HALT/LOAD RECOVERY

A problem in applying before images of audit records against ordered data sets has been corrected.

P1445 RECOVERY - STRETCH HEADER FOR CURRENT ROW

RECOVERY no longer writes to the last segment of the current row to stretch the header. The header will only be stretched if the next row needs to be allocated, in which case RECOVERY will write to the last segment of the next row.

P1583 RECOVERY - "ROLLBACK" USING TAPE AUDIT

ROLLBACK with audit on tape would fail with an "UNEXPECTED BLOCK SERIAL NUMBER" error just before the final phase of recovery. This has been corrected.

P1614 RECOVERY - RESTART RECONSTRUCTION AT RESTART POINT

Data recovery was sometimes unable to restart after creating a restart point; it failed with an "UNEXPECTED AUDIT RECORD TYPE" error soon after initiating the restart. This has been corrected.

P1621 RECOVERY - REBUILD FAILS WITH ON-LINE DUMP

Rebuild and reconstruct would fail with an "UNEXPECTED AUDIT RECORD TYPE" or INVALID INDEX if the dump used were taken on-line while RECOVERY was finishing. The problem has been corrected.

P1631 RECOVERY - "ROLLBACK" GETS TIMESTAMP MISMATCH

If a ROLLBACK were performed when Halt/Load recovery was pending, a timestamp mismatch error in the audit could be encountered. The mismatch no longer occurs.

P1632 RECOVERY - ERRONEOUS ROW LOCKOUT

When backing out certain types of audit records for certain structures, the recovery process could get false checksum errors. The recovery process must lock out any rows on which it gets I/O errors in order to proceed; consequently, rows were erroneously being locked out. This has been corrected.

P1633 RECOVERY - "RECOVERY" TAKES PROGRAMDUMP ON ERROR

RECOVERY now takes a programdump for abnormal termination.

P1634 RECOVERY - DATA RECOVERY SETTING BAD "EOF" POINTERS

It was possible for data recovery to incorrectly set the end-of-file pointers to the end of the last row of the file for partitioned structures. This has been corrected.

P1696 RECOVERY - "UNIDENTIFIED MISCELLANEOUS ERROR"

It was possible for RECOVERY to get an "UNIDENTIFIED MISCELLANEOUS ERROR" at 50325000 when processing an LGRA audit record as an after image. This has been corrected.

P1701 RECOVERY - AUDIT BEYOND "BLKIMG" RECORD FAILS

If there were any audit images for a locked out row after the BLKIMG audit record which caused the row to be locked, QUICKFIX would not apply them, but would still think it had fixed the row. (Such audit images could occur if several buffers for the row were in core at the time it was locked and they were subsequently modified, or if create/stores were being done beyond the end of file when the last row was locked out.) This could corrupt the data base. The error no longer occurs.

P1702 RECOVERY - "LIMIT = THRU AUDIT" FAILURE

If the input syntax to UTILITY for QUICKFIX for row lockout (recover rows using audit only) specified the prepass limit as "THRU AUDIT <integer>", RECOVERY would get a fatal error at the end of the prepass. This no longer occurs.

P8975 RECOVERY - HANDLE AUDIT BLOCK SIZE CHANGE

The ACCESSROUTINES were not handling a change in the audit block size made via DASDL UPDATE if the audit were on disk or pack. The audit could be corrupted as a result. The RECOVERY routines were not handling a change in the audit block size on an audit file switch,. These have been corrected.

P8976 RECOVERY - DIE ON "DBST" IF "REBUILD"

An abnormal termination no longer occurs on RECOVERY/<data base name> when a DBST record is encountered in the audit when applying before images in the cases where that can legally occur (i.e., a reconstruct with a finalaudit specified). This error was extremely unlikely to occur.

P8977 RECOVERY - POSITIONING THE AUDIT FOR RECONSTRUCT

When positioning the audit at the beginning of DATARECOVERY or RECOVERY while attempting to recover using an on-line dump, a fault could occur. This could only happen in the unusual case where there were no control records in the starting audit file which were written after the dump was started. The problem has been corrected.

P9017 RECOVERY - "NO FILE" "PARAMETERS"

An error in initialization for REBUILD, ROLLBACK and RECONSTRUCT has been corrected which caused a "NO FILE" on the parameters file if that file were located on pack instead of disk.

P9185 RECOVERY - ELIMINATION OF HOLES IN RESTART DATA SET

The ACCESSROUTINES no longer use records in the restart data set as temporary storage for last good restart areas. This results in less processor and I/O overhead, eliminates a potential source of fatal I/O errors in RECOVERY, simplifies the DMSII software, and can result in slightly less data being written to the audit trail.

It is very likely that rebuild would fail at sequence number 60933000 in RECOVERY. This has been corrected.

P9254 RECOVERY - CORRECT POSSIBLE "INVALID INDEX" IN "RECOVERY"

RECOVERY might have gotten an INVALID INDEX when processing audit records for ordered data sets. This has been corrected.

DMS II - REORGANIZATION

P1025 REORG - PROGRAM EFFICIENCY

Certain procedures in the REORGSYMBOLIC have been made more efficient.

P1055 REORG - CORRECT ALLOCATION OF RESTART FILE

REORGANIZATION will no longer ignore the INTERNAL FILES specification with respect to its restart files. Formerly, REORGANIZATION always allocated this file on disk.

P1063 REORG - "INVALID INDEX" REORGANIZING BIT VECTOR

Reorganization of a bit vector no longer terminates with an INVALID INDEX. Formerly, this would occasionally occur due to a miscalculation of an internal array size.

P1697 REORG - PROCESS TIME ACCELERATION

Local arrays used in iteratively-called procedures have been eliminated to reduce array allocation and deallocation time.

P1698 REORG - INVALID CHECKSUM

REORGANIZATION no longer generates an invalid checksum of block one of a disjoint ordered or unordered list.

P8912 REORG - SORT ERROR #4 ON INTERNAL SORT

REORGANIZATION will no longer terminate with a sort error #4 (disk limit specification exceeded) when sorting an internal file.

P8917 REORG - DISCONTINUITY RECORD ON FIXED STRUCTURE

REORGANIZATION now marks each structure which was "fixed" with an audit discontinuity. Formerly, only "generated" structures were marked.

P8918 REORG - RESEQUENCE DOCUMENT SECTION

The documentation section of DATABASE/REORGSYMBOLIC has been resequenced.

P9083 REORG - REMOVE USE OF CERTAIN COMPILE TIME IDENTIFIERS

All identifiers in the REORGSYMBOLIC using a peck mark followed by a node identifier have been changed to the peck mark number construct.

P9093 REORG - INVALID REORGANIZATION OF EMPTY SET

Reorganization of an empty disjoint set or subset no longer yields a new structure missing an "omega" entry.

P9183 REORG - INVALID REORGANIZATION OF RANDOM EMPTY BLOCKS

REORGANIZATION no longer terminates with an INVALID INDEX when the final reorganized structure contains two consecutive empty blocks in the basic scrambling area.

P9255 REORG - "NORECORD" ERROR EMPTY UNORDERED DATA CHAIN

If a master data set contained a valid address to an empty chain of unordered data set blocks to be reorganized, the REORGANIZATION would terminate with a NORECORD 5 error.

DMS II - UTILITY

D2243 UTIL - SPECIFY "PACKNAME"

The ability to specify PACKNAME has been added to the row selection criteria for the DUMP, COPY and RECOVER statements. By using PACKNAME in conjunction with FAMILYINDEX, it is now possible to limit UTILITY to a particular pack without having to enumerate the files that exist on that pack.

When more than one selector is specified in the rowselection criteria, they are now separated by either "AND" or "&", rather than separated by a comma. The old syntax will result in a syntax error.

If more than one integer or range of integers (separated by commas) is given for the family index or row options, the conditions are OR'ed together. If any one of the ranges or integers are correct, the selector is satisfied.

Examples:

1. DUMP=(FAMILYINDEX=1,4 AND PACKNAME=DBDATA) TO TAPEX

   All rows in the data base that reside on pack family DBDATA AND are on either familyindex 1 OR 4 of that family will be dumped to TAPEX.

2. DUMP (DB/A/=, DB/B/=, DB/RDS/= (PACKNAME=DBDATA))
   (PACKNAME=DBDATA&FAMILYINDEX=1) TO TAPEX

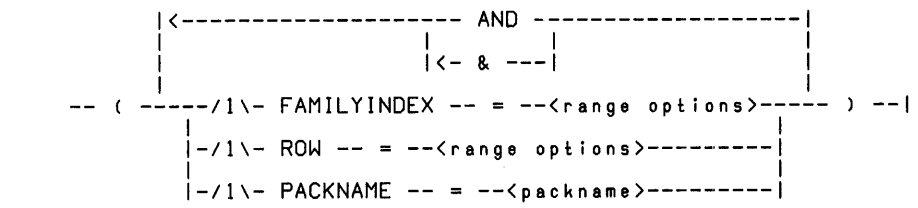
   All rows in files DB/A/= and DB/B/= which are on familyindex 1 of DBDATA will be dumped. The rows of DB/RDS/= will be dumped if they satisfy either the inner condition packname=DBDATA or the outer condition that packname is DBDATA and FAMILYINDEX=1. Since the first condition is less restrictive, all the rows of DB/RDS/= on DBDATA will meet it and be dumped.

3. RECOVER(ROWS USING BACKUP) = (ROWLOCK=READERROR, LOCKEDROW AND
   PACKNAME=DBDATA AND FAMILYINDEX=1-3,5) FROM TAPEX

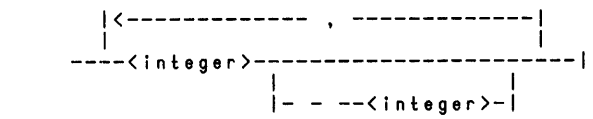
   If there are 5 packs in the DBDATA family numbered 1-5, none of the rows on familyindex 4 will be selected for reconstruction. All of the rows of familyindex 1,2,3 or 5 that either have a READERROR or are LOCKED OUT will be reconstructed.

The DUMP syntax is as follows:

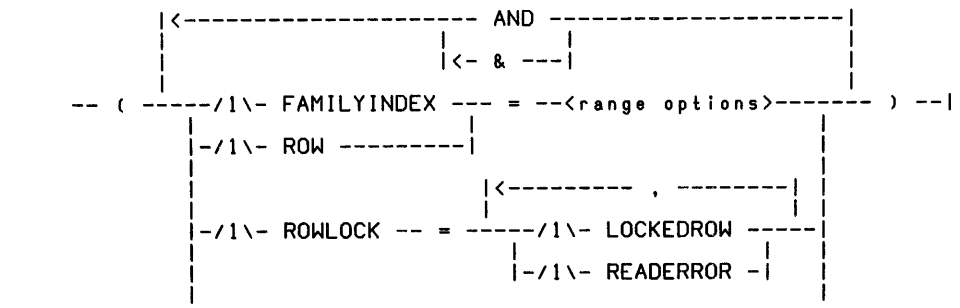
<rowselector>



<range options>



The RECOVER syntax is as follows:



```
|-/1\- PACKNAME -- = --<packname>------------|
```

D2441 UTIL - MULTIPLE WORKER RECOVERY

The ability to specify how many workers in parallel will be processing tapes on input has been added to UTILITY. This capability is available for UTILITY functions which use tapes as input; i.e., RECONSTRUCT, REBUILD, and COPY. The complete syntax follows:

RECOVER Statement

```
------------------------------ RECOVER --<recover specification>----->
  |                            |
  |- OPTIONS --<option list>-|

  |<----------- ; -----------|
  |                          |
>---<filelist>--<sourcelist>-----------------------------------------|
```

<option list>

```
       |<-------------------- , -------------------|
       |                                           |
-- ( -----/1\- NOZIP ------------------------------- ) ---------------|
        |                                        |
        |-/1\- FLUSHDB -- = --<integer>---------|
        |                              |        |
        |                              |- MIN -|
        |                                        |
        |-/1\- WORKERS -- = --<integer>---------|
```

<recover specification>

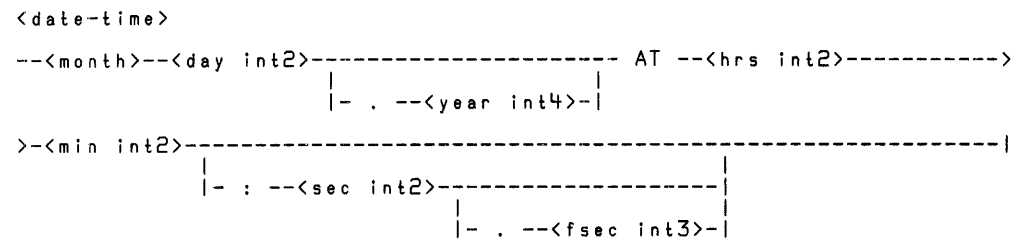
```
-- ( --- ROWS USING BACKUP --------------------- ) ------------------|
      |                                       |
      |- ROWS IN PLACE USING BACKUP ----------|
      |                                       |
      |- ROWS USING AUDIT ONLY -- , --<limit>-|
      |                                       |
      |- ROLLBACK ---<point>------------------|
      |            |
      |- REBUILD --|
```

<limit>

```
-- LIMIT -- = ---<integer>--- CONTROL --- POINTS ---------------------|
               |           |           |            |
               |           |- SYNC ----|            |
               |                                    |
               |- THRU AUDIT --<integer>-----------|
               |                                    |
               |-<date-time>-----------------------|
```

<point>

```
---- THRU AUDIT --<integer>-------------------------------------------|
  |                                                                 |
  |- TO --- BOJ --- OF -- <int4> / <int 4> ---------------------|
        |        |                          |                      |
        |- EOJ -|                          |- ON <date-time> -|
        |                                                          |
        |- GEQ ---<date-time>------------------------------------|
        |        |
        |- LEQ -|
```



<month>

JAN, JANUARY, FEB, FEBRUARY, etc.

<day int2>

DAY = Unsigned integer of not more than 2 digits.

<year int4>

YEAR = Unsigned integer of not more than 4 digits.

<hrs int2>

HRS = Unsigned integer of not more than 2 digits.

<min int2>

MIN = Unsigned integer of not more than 2 digits.

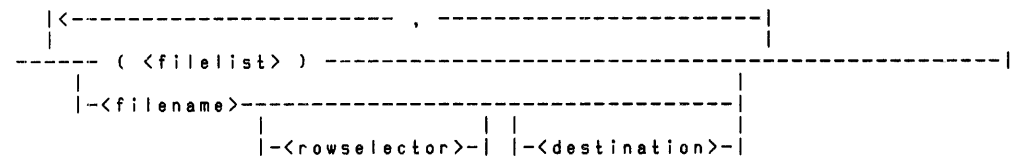
<sec int2>

SEC = Unsigned integer of not more than 2 digits.

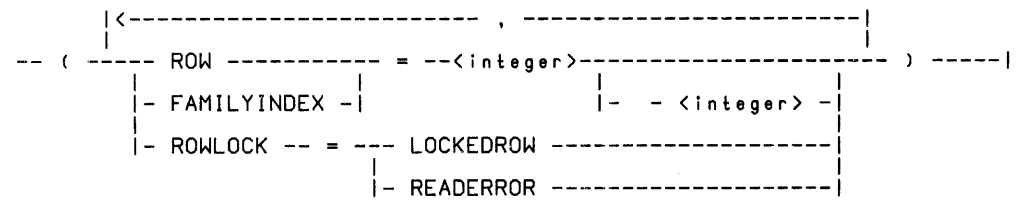
<fsec int3>

FSEC = Unsigned integer of not more than 3 digits.

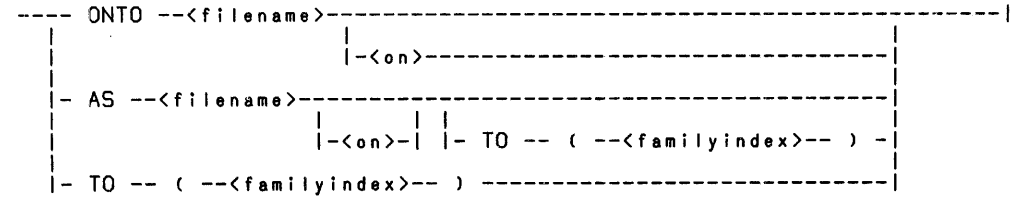
<filelist>



<rowselector>



<destination>



<on>

```
-- ON --<familyname>--------------------------------------------------|
```

<familyindex>

```
-- FAMILYINDEX -- = ---<integer>---------------------------------------|
                     |           |
                     |- RETAIN --|
```

<sourcelist>

```
-- FROM ---------------------------------------------------------------->

   |<---------------------------- , ------------------------------|
   |                                                              |
   |                 |<------------------ , ------------------|   |
   |                 |                                        |   |
>---<tapename>-- ( ----- VERSION -- = --<integer>------------ ) ------|
                       |                                    |
                       |- CYCLE -- = --<integer>----------|
                       |                                    |
                       |- SERIALNO -- = ---<integer>------|
                                          |                 |
                                          |- " <string> " -|
```

Restrictions:

Certain forms of the RECOVER statement require both a <filelist> and a <sourcelist>; others do not. The following table indicates the requirements of the various forms of the RECOVER statement.

Statement	Filelist	Sourcelist
RECOVER(ROWS USING BACKUP)	Required *	Required
RECOVER(ROWS IN PLACE USING BACKUP)	Required **	Required
RECOVER(ROWS USING AUDIT ONLY----)	Not Permitted	Not Permitted
RECOVER(ROLLBACK----)	Not Permitted	Not Permitted
RECOVER(REBUILD----)	Not Required ***	Required

* If "ONTO <filename>" or "AS <filename>" is used, the <filename> must not be the same as any data base file name in this or any other data base.

** <destination> is not permitted.

*** "ONTO <filename>" and "AS <filename>" are not permitted.

COPY Statement

```
---------------------------------------------------- COPY ----------->
  |                                                |
  |- OPTIONS -- ( -- WORKERS -- = --<integer>-- ) -|

  |<----------- ; -----------|
  |                          |
>---<filelist>--<sourcelist>------------------------------------------|
```

Examples:

1. OPTIONS(NOZIP) RECOVER(ROWS USING BACKUP) = FROM
   T1,T2,T3,T4,T5,T6,T7,T8

Since the example does not use the WORKERS=<integer> construct, UTILITY will process each of the specified tapes serially, and process the cycles of each tape in parallel. In order to achieve a completely serial loading of the tapes, use the construct " WORKERS = 1 " in the RECOVER statement.

2. OPTIONS(NOZIP,WORKERS=3) RECOVER(ROWS USING BACKUP) =
   FROM T1,T2,T3,T4,T5,T6,T7,T8

The new syntax informs UTILITY to process three tapes at a time. UTILITY starts with the first three tapes in the tapelist and fires off a separate worker to process each one. Each tapeid could have multiple cycles, and the tape list of 8 tapes could actually represent more than 8 physical tapes to be processed. The worker mechanism ensures a total of 3 tapes will be processed at a time, whether they are different cycles of one tapeid or different tapeids. UTILITY will continue processing 3 at a time until the tape list is exhausted.

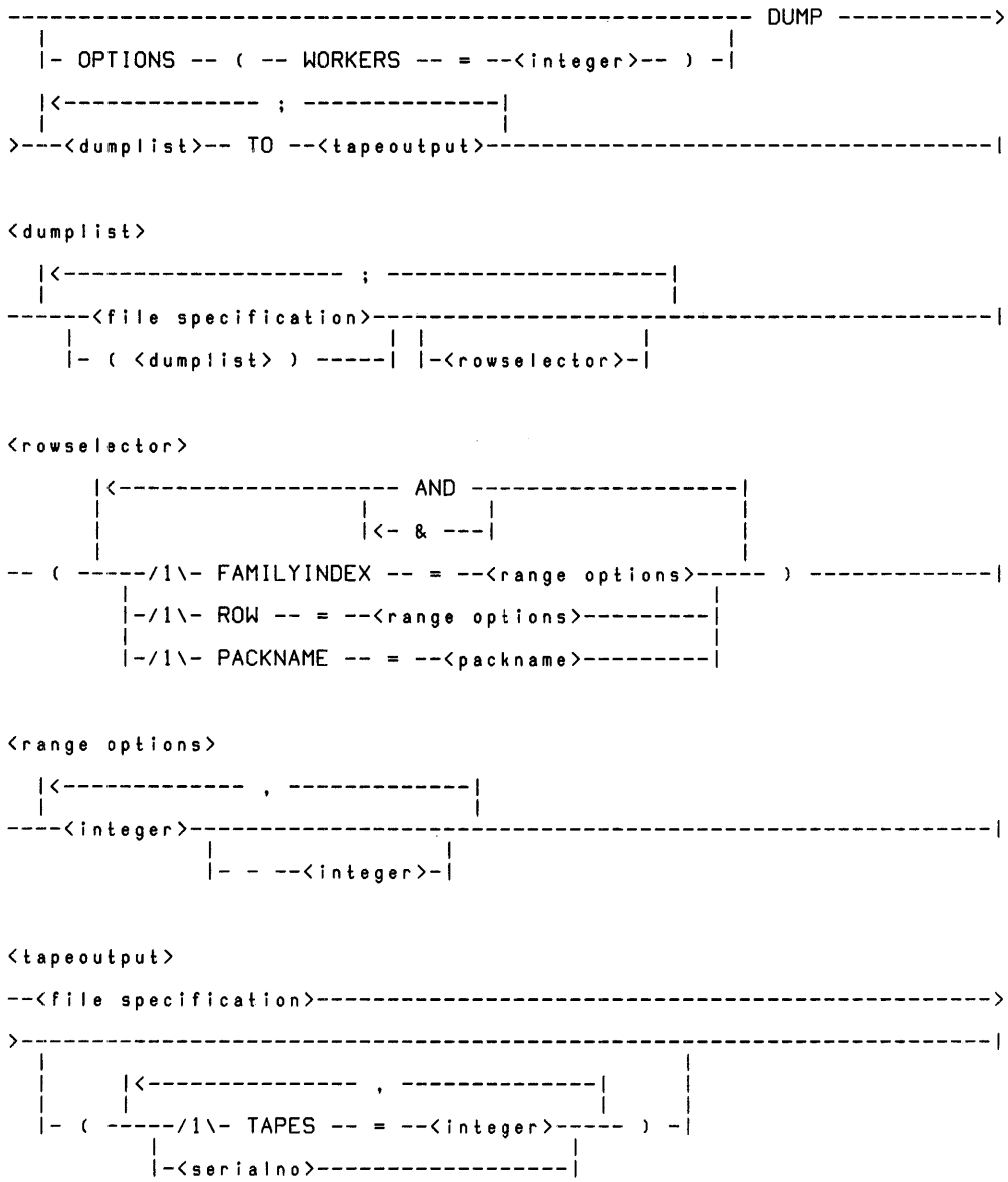
When the WORKERS construct is used, the number of cycles of a tape does not determine the amount of parallel processing that occurs during UTILITY recovery. The parallelism on tape input is solely determined by the number of workers specified.

Multiple Worker Dumps

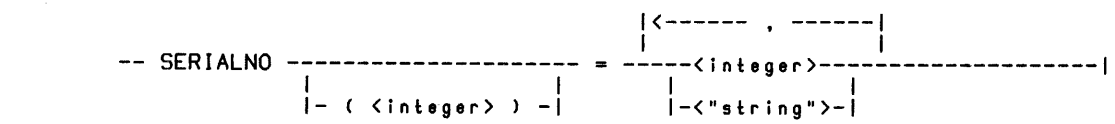
The parallel processing concept will also be available for a dump but in a different form than before. The semantics of the TAPES=N construct has been changed and the WORKERS=N construct added.

By using a multiple workers construct, the numbers of tapes dumped in parallel can be determined, independent of the number of cycles of a given tape. The ability to specify multiple dumplists and tapeids in one dump statement has been added so that multiple UTILITY dumps do not have to be run in order to logically partition the files to be dumped.

DUMP Statement



<serialno>



Examples:

1. DUMP = TO TAPEX (TAPES=3)

The meaning of the TAPES=<integer> construct will remain the same when not used in conjunction with the WORKERS construct. The TAPES construct indicates that the directory of rows to be dumped wil be divided into three parts, which will be dumped in parallel. In this example the tapeids will be TAPEX, CYCLE=1; TAPEX, CYCLE=2, and TAPEX, CYCLE=3.

2. DUMP DB/A TO TAPE1, DB/B TO TAPE2, DB/C TO TAPE3

In this example, the user is logically partitioning the files to be dumped into three parts rather than letting UTILITY do it. Each part will be assigned to a tape with a unique tapeid and will have a completely different directory.

3. OPTIONS(WORKERS=3) DUMP = (FAMILYINDEX=1) TO T1(TAPES=3),
= (FAMILYINDEX = 2) TO T2(TAPES=3), = (FAMILYINDEX=3) TO
T3(TAPES=3)

In this example the user is dumping each unit of a pack family to a separate set of tapes. The rows on each pack will be divided up by UTILITY into three parts, which will be dumped to different cycles of the same tape name. We have a total of nine tapes which could be processed simultaneously, but the WORKERS=3 will limit us to processing three tapes at a time.

4. OPTIONS(WORKERS=3) DUMP
= (FAMILYINDEX=1) TO T1(TAPES=3,SERIALNO(1)=100,101,
SERIALNO(2)=200,201 , SERIALNO(3)=300,301) ,
= (FAMILYINDEX=2) TO T2(TAPES=3,SERIALNO(1)=110,111 ,
SERIALNO(2)=210,211 , SERIALNO(3)=310,311) ,
= (FAMILYINDEX=3) TO T3(TAPES=3,SERIALNO(1)=120,121 ,
SERIALNO(2)=220,221 , SERIALNO(3)=320,321)

This example is identical to the previous one except that the tapes have been assigned serial numbers. SERIALNO(1) means the first cycle of that tape name. The first reel of cycle 1 of T1 will be assigned serial number 100. If it overflows, the second reel (version 2) of cycle 1 will be assigned number 101, etc. This syntax is not new, but is probably misunderstood.

Operator Interfaces to UTILITY

The number of workers specified in the input can be changed during the execution of UTILITY by entering an accept message telling it the new value for workers. The syntax for the message is:

<mixno> AX WORKERS = <integer>

This can be entered even if the WORKERS construct was not used in the original input. The mixno given must correspond to the main UTILITY stack.

Restart Request

The ability to restart UTILITY after it has been DSed has been added. This capability is available for UTILITY dumps and RECOVERY runs that have begun processing tapes and are DSed either internally (due to fatal I/O errors) or by the operator. The restart request will cause UTILITY to start over on the tape that was being processed at the time of the DS, without having to reprocess any tapes that were completed. The input string to UTILITY is not scanned on a restart request, and can be left blank. The mix number of the UTILITY run to be restarted is passed to UTILITY via the taskvalue attribute, and is negated to distinguish it as being a restart mix number. The following illustrates a restart request:

RUN UTILITY/TEST (" "); VALUE = -1025

UTILITY restarts itself by using the HLDUMPINFO file that it creates with the mix number as part of the title. If the file TEST/HLDUMPINFO/1025 does not exist (has been removed), UTILITY will not restart. In addition each tape or dump worker that was active at the time of the DS created its own HLDUMPINFO file that must be present in order for the worker to restart where it left off. These files are identified by the mix number and a worker number; e.g., TEST/HLDUMPINFO/1025/01.

Continued Request

It is now possible to specify a continued UTILITY request for REBUILD or RECONSTRUCT. If the original request did not load the complete set of files and/or rows due to I/O errors or user oversight in specifying the tapelist, a special continued request will allow the missing files and rows to be added before firing off RECOVERY.

A taskvalue of 8 identifies a continued UTILITY request, and the input syntax is the same as a regular UTILITY request. The user must make sure to use the same type of REBUILD or RECONSTRUCT request, while changing the filelist and/or tapelist to add the necessary files. UTILITY will use the RECONSTRUCTINFO or REBUILDINFO file created by the previous UTILITY run, and simply add to it. This file must be present and be the correct one.

For example, a user, in trying to do a REBUILD, forgot to specify one of the required tapes in the tapelist. Assuming that the database was dumped to four tapes; T1, T2, T3 and T4, the UTILITY input might be the following:

```
RUN UTILITY/TEST ("RECOVER(REBUILD THRU AUDIT 5) FROM T1,T2,T4 ")
```

Since a REBUILD must load the entire database, UTILITY would notice that some of the files were not loaded and stop the REBUILD. Previously, the UTILITY REBUILD would have to be completely re-run, although most of the files were successfully loaded. But using the continued request, the user would just have to specify the missing tape in the tapelist and none of the other tapes would have to be reloaded; e.g.,

```
RUN UTILITY/TEST ("RECOVER(REBUILD THRU AUDIT 5) FROM T3 ");
VALUE=8
```

I/O Errors During Dump

UTILITY will handle both hard I/O errors and timeouts that occur while writing to tape. Appropriate error messages that describe the type of error that has occurred are displayed. The tape on which the error occurred is labeled "BADTAPE" and closed. The following accept message is then displayed to the operator:

```
AX "OK" FOR RETRY OR NEWTAPE
```

At this point the operator can either enter an "AX OK" or DS the job. An "OK" will cause UTILITY to restart the dump at the same point where the badtape was started. At this point the operator can switch tapes, clean the tape that got the error, change tape drives, etc.

Other UTILITY workers can proceed with their dumping while this worker is waiting for the accept message or waiting for the tape to be put up.

I/O errors that occur while UTILITY is reading from disk are fatal. UTILITY cannot successfully complete a dump unless all the rows that are specified in the dumplist are dumped. The row that caused the error on disk must be reconstructed before it can be dumped. UTILITY will DS itself in this case, leaving its restart files around so that it can be restarted and continue dumping where it left off.

I/O Errors During RECOVERY

If UTILITY encounters hard I/O errors or timeouts while reading from tape, the appropriate error message is printed and the following is displayed to the operator:

```
AX 'RETRY' OR 'SKIP ROW' OR 'QUIT TAPE'
```

If the operator enters "RETRY", UTILITY will start over at the beginning of the tape, and attempt to read it again. "SKIP ROW" causes the row on the tape that got the I/O error to be passed over and not be included in the RECOVERY operation. If UTILITY skips a row in this manner, RECOVERY will not automatically be fired off so that the user can have the opportunity to load the missing row(s) via a UTILITY continued request before initiating RECOVERY.

"QUIT TAPE" will cause the rest of the rows on the tape (and any subsequent reels) to be skipped. If there are other tapes in the tape list to recover from, UTILITY will proceed to load those tapes. As with the "SKIP" response, RECOVERY will not automatically be fired off so that the missing rows can be loaded from other tapes.

D2482 UTIL - NO FILE ON "CONTROLOLD"

Under certain circumstances, the control file was not being read off the tape used for recovery; consequently, UTILITY would get a NO FILE on CONTROLOLD. This has been corrected.

D2497 UTIL - NEW METHOD OF DATA BASE INITIALIZATION

CURRENT METHOD FOR INITIALIZATION

Currently, initialization of DMSII data base files is accomplished by the execution of an OPEN INITIALIZE statement in a host language program. This action may be performed at any time when no other activity on the same data base is occurring. The effect is to initialize all structures which were invoked. Since it is not possible to completely control the invocation of structures in a user program, there are a number of undesirable side effects to this procedure:

1. Global data can only be avoided by using logical data bases which do not include it. It may be necessary to have some special logical data bases for just this purpose.

2. The restart data set must be invoked in order to do any updates of an audited data base.

3. Special handling in the accessroutines is required for the partition directory dataset which gets invoked along with any partitioned structure.

4. Embedded structures can only be invoked by invoking their masters.

In any of these cases, structures can be initialized accidentally or when it is not desired to do so (or when it is disastrous to do so).

With the advent of the control file, it is possible to improve upon this situation. The control file serves as a central repository for data base state information and can be used as a data base lock. In both of these functions, it can help improve the initialization process.

NEW METHOD FOR INITIALIZATION

1. DATABASE/UTILITY has a new section which provides an initialization function. Run as a separate program from the accessroutines, it takes as input a string (possibly on card images) of the form

   "INITIALIZE <structure list>;"

   The <structure list> can be a list of structure names, separated by commas, or an equals sign (=), indicating the entire data base. (The equals sign now may also be used in LIST or WRITE commands to UTILITY when it is desired to print the entire data base. The former use of "ALL" for this print function is maintained.)

   UTILITY checks the list of structures for "consistency" and then create new "empty" files in their place. The control file file-state fields are adjusted to reflect the initialization (CFAUDINZ for audited data bases, CFFILENORMAL for unaudited).

   "Consistency" is defined as a list of structures which leaves no "dangling pointers"; i.e., set entries, roots, or links left pointing into the to-be-empty file. Only disjoint structures may be named, since embedded structures only would necessarily leave "dangling roots". Sets of, and embedded structures within, the named structures are included by default. Disjoint manual sets can be named, but not automatic ones if the data set is not also named. If structures with populations in global data are named, all such structures and global data must be named. Structures which are the object of counted links in a named structure must also be named.

   This function, like the current OPEN INITIALIZE, is an exclusive function on the data base and has a control file lock. It needs to complete before releasing the data base. It can be executed in several steps - doing different structures each time. It is also repeatable in case it is run incorrectly or it is interrupted (e.g., by a Halt/Load).

2. DASDL has an option to prohibit OPEN INITIALIZE on a data base. The syntax for this option is

   NO OPEN INITIALIZE

   Use of this option gives a data base administrator more control over the initialization of the data base. The initialization can only be done with UTILITY, a program more easily controlled by the d.b.a. than arbitrary host-language programs. An attempted OPEN INITIALIZE on a data base with this option set will result in a (DM)OPENERROR 43 exception.

3. For the present time, OPEN INITIALIZE will remain unchanged, for reasons of compatibility.

DMS II - UTILITY

P1072 UTIL - FILE LISTED INSIDE PARENTHESES

UTILITY was incorrectly parsing a dump list in which the files were listed inside parentheses and no fileselector list followed. This has been corrected.

P1210 UTIL - CORRECT BLOCKSIZE FOR "CFOLD"

The blocksize used for reading in the old control file from tape was inconsistent with the blocksize used by UTILITY. This has been corrected.

P1211 UTIL - LABEL "BADTAPE"

If UTILITY encounters an error when creating a tape, that tape should be labeled as "BADTAPE". This failed to occur if the error was encountered when verifying the tape (i.e., reading it). This has been corrected.

P1212 UTIL - CORRECT ALLOCATION OF TAPES

Previously, if a large number of tapes were specified and a very small number of rows were dumped, incorrect tape dumping could occur. UTILITY will now allocate tapes so at least one row is included on each tape. This may cause the number of tapes used to be fewer than the number of tapes specified.

P1278 UTIL - RECOVERY USING "II.8" DUMP TAPES

The control file that is created by UTILITY when attempting to reconstruct or rebuild from a II.8 dump tape was causing problems under certain conditions. These problems have been corrected.

P1355 UTIL - REMOVE FIRST READ ERROR MESSAGE

The timestamp that told UTILITY when the first read error occurred had been deimplemented; however, the message was still being displayed. This has been corrected.

P1356 UTIL - FAULT INITIALIZING LARGE NAME

UTILITY no longer faults with a SEG ARRAY error when initializing structures from a data base when the data base name contains more than nine characters.

P1581 UTIL - HALT/LOAD RESTART WITH MULTIPLE CYCLES

UTILITY was failing to restart correctly following a Halt/Load if some of the cycle of a multiple cycle dump were already finished while others had not yet started. UTILITY has been corrected so that cycles that had not yet begun will start again at the beginning.

P1597 UTIL - TAPE KIND SET CORRECTLY FOR FILE ON PACK

If a user does a UTILITY dump to a file on PACK, the KIND will now be set correctly so that a "REQUIRES DK DISK" message does not appear.

P1622 UTIL - ROW SELECTION WITH MULTIPLE OPTIONS

In some cases UTILITY was not behaving correctly when multiple row selection options were given. This has been corrected.

P1629 UTIL - SYSTEM PARTITIONED FILES CAN BE DUMPED

UTILITY was unable to dump partitioned file that had no usercode since GETSTATUS returns an asterisk as part of the filename name. The asterisk returned in this case will now be ignored.

P1630 UTIL - FAULT IN "REALSWITCH"

A problem in procedure REALSWITCH which caused a fatal error during a COPY has been corrected. The problem occurred because (1) OPENTAPE was not invoked to set up the proper variables and open the required tape file, and (2) SKIPBLOCKS was invoked with the wrong parameter causing a fatal error from ASSERT.

P1635 UTIL - "PRINTIT" GIVES ERROR FOR BLOCK ARRAY OVERFLOW

PRINTIT will now give an error if it is about to overflow its NOTAVAIL block array, rather than give an INVALID INDEX. The size of the array has been increased also.

P1699 UTIL - INITIALIZING STANDARD DATA SETS

A problem has been corrected in INITIALIZING standard data sets which have less than three segments per area.

P8944 UTIL - NEW TAPE BLOCK DEFAULT SIZE

The tape block size default value has been changed from 454 to 903 on III.0 UTILITY. II.9 UTILITY will accept either block size on a dump tape, although it will continue to dump tapes with a block size of 454.

P8985 UTIL - RECONSTRUCT ONTO "DB" FILES

UTILITY will now check the file kind of the temporary files that RECONSTRUCT is using. If the filekind is DBDATA or DBRESTARTSET, UTILITY will issue an error and stop. This is to prevent the accidental use of an existing data base file as a temporary reconstruct file, thereby wiping out a data base file. This check will be made for reconstruct "ONTO" and "AS". The temporary files for a reconstruct "AS" will be given filekind DATA.

P9019 UTIL - RECONSTRUCT EMPTY FILE

An empty direct data set can now be reconstructed without getting a syntax error.

P9076 UTIL - NO RECONSTRUCTION ON ROW NOT DUMPED

UTILITY was incorrectly initiating reconstruction on rows which were present at dump time but which were not dumped to the tape being used for reconstruction. This has been corrected.

P9182 UTIL - NO FILE FOR DUMPS TO DISK

UTILITY was locking the dump file only if KIND were PACK, so that dumps to disk were not saved. This has been corrected.

P9184 UTIL - NEW "III.0" FORMAT TAPE BLOCKS

Due to a change in the format of a tape block that will be released on the III.0 release, a modification was necessary to II.9 so that it will be able to handle either the old or new format. The change involves the addition of two more control words at the beginning of each tape block.

P9258 UTIL - "READERROR" SET FOR SELECTED ROWS

UTILITY reconstruct was marking all the rows of a file to be reconstructed with a READERROR rather than only those rows which were actually to be selected for reconstruction. This has been corrected.

P9278 UTIL - PROBLEMS WITH PARTITIONS

Calls on the control file module to check time stamps were being done incorrectly for partitioned structures. Partitioned structures are now using a different set of control file procedures, passing the partition numbers and structure number.

P9279 UTIL - PROBLEMS IN "RECONINFO"

The procedure that creates the entires in the RECONINFO file was incorrectly handling a situation where it should delete an entry and insert a new one. This resulted in the possibility of having a blank record in the RECONINFO file, which will cause recovery to fail.

DUMPALL

P9031 DUMPALL - SUPPRESS EXTRA LINES IN "LAN" LIST

DUMPALL, using the LAN option, no longer prints two EBCDIC lines on continuation lines of records which are larger than 56 characters.

P9113 DUMPALL - PACKNAME OPTION FOR FILE ROUTINE

DUMPALL no longer terminates with an attribute error for the following input:

FILE <filename> PACK = <packname>

P9114 DUMPALL - CORRECTLY PRINT VARIABLE RECORDS

DUMPALL now blank fills after variable length records before printing them for the LIST and COPY routines. Variable length records of disk and pack files, which have FILETYPE=3, cannot be blank filled since the logical record size is not known.

P9213 DUMPALL - CORRECT PACK HANDLING

DUMPALL will no longer hang on "NO FILE" when attempting to list a non-present file, using the "PACK" option.

P9214 DUMPALL - PROPAGATE SAVE FACTOR

The save factor of DUMPALL's input file is now carried over to the output file when appropriate.

P9215 DUMPALL - REVISE TAPE HANDLING IN "DMPMT"

The way in which tape files are handled for the DMPMT option has been changed.

Record and block sizes are now determined from the file's header record, where possible, instead of being static.

ESPOL

P1390 ESPOL - BINDINFO FOR DOUBLE PRECISION VARIABLES

Correct bindinfo is now generated for double precision variables.

ESPOL INTRINSICS

P1156 ESPOLINTRN - "TAN" AND "COTAN" INTRINSICS

The TAN and COTAN intrinsics will now return the maximum single precision value as an approximation for infinity when the argument is appropriate (for TAN, argument MOD PI=PI/2; for COTAN, argument MOD PI=0). Previously, this condition was not correctly recognized.

P1157 ESPOLINTRN - "DGAMMA" INTRINSIC

An error termination is now obtained when a zero argument is passed to the DGAMMA intrinsic.

P1362 ESPOLINTRN - "BASIC," MAKING ARRAYS PRESENT VIA "TOUCH"

The BASIC intrinsics will make arrays present (when necessary) via the proper invocation of the TOUCH intrinsic.

P1420 ESPOLINTRN - "ALGOL" OUTPUT ARRAY ELEMENT

The problem of an ALGOL program causing a fault in the intrinsics when attempting to output an array element using a */ or *// format has been corrected.

P1421 ESPOLINTRN - "BASIC," "SETW" IMPROVEMENTS

The SETW statement of BASIC will now work correctly for the following cases:

1. SETW <filenumber> TO x, where 80 < x <= 160.

2. SETW <file number> TO x, where the length of the file (LFW) is not a multiple of 80 and LFW-(LFW MOD 80) < x <= LFW.

P9115 ESPOLINTRN - "REPLACE FOR NUMERIC" CORRECTION

The following statement in ALGOL was producing the resultant string of "1" right-justified in a field of <number> characters, when <expression> was a floating-point value with <number> significant digits to the left of the decimal point:

REPLACE <pointer> BY <expression> FOR <number> NUMERIC

The OUTPUTCONVERT intrinsic will now produce the correct resultant string in this case.

P9116 ESPOLINTRN - "ATTRIBSEARCHER" CHECKING MNEMONICS FOR "PLI"

The ATTRIBSEARCHER intrinsic will now properly recognize that run-time errors converting a string mnemonic to a value for PL/I programs are user errors and not compiler errors.

FILECOPY

D2225 FILECOPY - "EXCLUDE" SYNTAX CORRECTION

The EXCLUDE syntax has been corrected to allow "*X" to compare equally with "X".

D2258 FILECOPY - "NO FILE" ON "INCLUDE" STATEMENT

FILECOPY will now display the non-existent file message to a NO-FILE response from GETSTATUS on a file title in an INCLUDE statement. This will cause the task to continue and not be aborted. The error message "UNKNOWN GETSTATUS ERROR" has been changed to read "GETSTATUS ERROR NUMBER XXX", where XXX is the GETSTATUS run number.

D2272 FILECOPY - SAVEFACTOR ZERO VS "FILECOPY"

A new modifier SFACTORZERO has been added to FILECOPY. This modifier changes the way FILECOPY handles files with SAVEFACTORS of ZERO when the EXPIRED task option is used. If the modifier is not used, the zero savefactor is used to determine if the file meets the request criteria. If the modifier is used, the SAVECONSTANT value is used in place of the SAVEFACTOR.

Syntax:

```
-- SFACTORZERO --|
   -----
```

The minimum abbreviation is underlined in the above.

D2485 FILECOPY - "SYNTAX" MODIFIER

A new modifier has been added to FILECOPY to aid in request checking. The modifier, SYNTAX, causes the FILECOPY run to only do the input syntaxing and nothing else. It should be noted that the use of SYNTAX in a task of a multi-task FILECOPY run will cause all tasks to stop after the input syntax checking is complete.

FILECOPY

P1042 FILECOPY - "(<USERCODE>) =" SYNTAX ERROR

FILECOPY will give a syntax error for the following input syntax:

```
(<usercode>) =
```

The following input syntax will work correctly:

```
(<usercode>)=
```

P1289 FILECOPY - "FKINDS" ARRAY ELIMINATED

The value array FKINDS has been replaced by the ATTRIBSEARCHER intrinsic; consequently, FILECOPY can now automatically handle new FILEKINDS.

P1290 FILECOPY - "B6800" VS. "FILECOPY" HEADING

FILECOPY will now properly display its report heading when run on a system with a 4-digit system identifier (B6800).

P1291 FILECOPY - LARGE "INCLUDE" AND "EXCLUDE LIST"

FILECOPY will now handle large INCLUDE and EXCLUDE lists for a single family. When enough input had been read to cause the creation of the AUXFILE from the AUXBUF, the record locater for FAMILY INFO was not reset to the proper record. Also, most of the fields in the FAMILY INFO record and AUXWORD have been expanded to handle very large requests.

P1337 FILECOPY - CARD INPUT VS. "UNITS=1"

FILECOPY will now give a syntax error when the INPUT CARD file has UNITS set to 1 (CHARACTERS).

P1477 FILECOPY - LOOP ON INPUT

A possible loop on scanning input data has been corrected.

P1508 FILECOPY - MULTIPLE TASK EXECUTION

FILECOPY will now handle multi-task requests without terminating with a SEG ARRAY error in RETURNATOKEN.

P9000 FILECOPY - SUPERFLUOUS COMMA IN "WFL" DECK

FILECOPY will now create correct WFL output decks when identical tasks are requested in the same run.

P9001 FILECOPY - ADDED TASK EXCLUDE INDEX FILE

The INDEX file will now be created and used correctly by FILECOPY when using the ADDED task request. All files in the index will be excluded (unless overridden by an INCLUDE statement).

P9189 FILECOPY - EXPIRED TASK

FILECOPY will now handle the expired task correctly when a file has a LASTACCESSDATE of a year older than run date.

P9191 FILECOPY - "EXTRACTONEFILE" VS. "GETSTATUS" ARRAY

FILECOPY will now handle the case when a continuation GETSTATUS call is requested by EXTRACTONEFILE procedure, and a complete file name and information have been set up (GOTAFILE = TRUE) from the previous array. Previously, this caused a file that met the requested criteria not to be copied.

P9192 FILECOPY - SUMMARY OUTPUT REFORMAT

The summary portion of the FILECOPY output has been reformatted to remove the possibility of SEG ARRAY errors.

P9289 FILECOPY - SCANNER FIELD EXTENSIONS

FILECOPY will now allow almost unlimited number of input records and tokens, as all scanner fields have been expanded to handle 65,000 entries.

P9290 FILECOPY - "AUXFILE" AND "AUXBUF" FIELD EXTENSIONS

FILECOPY will no longer stop on a "NO FILE AUXFILE" when handling larger amounts of input.

FILEDATA

P1185 FILEDATA - "LFILES CAT"

FILEDATA now gives catalog information to the CANDE user when "LFILES:CAT" is entered.

P1186 FILEDATA - INPUT STRING >255 CHARACTERS

FILEDATA can now handle input string greater than 255 characters.

P1339 FILEDATA - DETERMINE SYSTEM TYPE

The system type in now used properly when printing headings.

P1392 FILEDATA - RUNNING ON "B6800"

FILEDATA now recognizes machines types other than B6700; i.e., B6800. This information is used for headings on output.

P1659 FILEDATA - "TD830" VS. NON-TD830 "ODT"

The extra ETX character has been eliminated from systems which are using TD800(TD804), CONRAC as ODTs. The ETXs are required on TD830 ODTs by the terminals' internal firmware.

FILEDATA has also been corrected; it must be used in conjunction with this change.

The following hardware must be changed in some cases to cause the correct terminal type to be reported:

B6700 with SLC ODT controls:

CONRAC	The terminal ID block on the SLC needs a pin inserted in the "CON" location of each ODT.
TD800	The terminal ID block on the SLC needs a pin inserted in the "9348" location of each ODT.
TD830	The terminal ID block on the SLC needs a pin inserted in the "9348" and "BIDS" location of each ODT.

B6800 with SDC II ODT controls:

TD830	These are the only ODTs connected to this control. They require no change for the use of these patches.

FORTRAN

D2402 FORTRAN - LOWER BOUNDS FOR "FORTRAN" ARRAYS

Lower bounds for arrays have been implemented in FORTRAN, providing the ability to specify both lower and upper bounds for each dimension of an array.

As a result of this implementation, some problems with the previous compiler have been corrected. Subscripts will now be printed correctly when an array with variable bounds is output using free-formatting and named output is requested (e.g., using a free-format designator of */). Some inconsistencies have been eliminated between the OPT=0 and the OPT=1 compilers in calculating linearized subscript values from subscripts or in calculating array sizes for I/O when variable bounds or individual subscript expressions are not of type integer, so OPT=0 and OPT=1 will produce the same results. With the exception of these problem areas, existing FORTRAN programs will require no conversion and will produce the same results when compiled with the new compiler as they do when compiled with the previous compiler.

The following changes should be made to the FORTRAN Manual to document this new implementation.

1. To the list of "Basic FORTRAN Characters" for EBCDIC input in Chapter 3, add the character ":" (colon).

2. In Chapter 5, replace the section on "Array Elements" by the following:

ARRAY ELEMENTS

FORTRAN variables may be divided into simple variables, which are denoted by a variable name only, and array elements. An array element is a member of a data structure called an array.

An array is an ordered set of data organized into dimensions. An array is referenced by an array name which is of the same form as a variable name. A variable name is designated as an array name by its appearance in an array declarator (see Chapter 7). All of the elements of an array are of the same type, and this type is indicated by the array name in the same manner as for variable names.

Each member of an array is an array element. The proper format for an array element is:

```
+-----------------------------------------+
|                  a(s)                   |
|                                         |
+-----------------------------------------+
|      where a is an array name,          |
|        (s) is the subscript,            |
|     and s is a list of subscript        |
|   expressions separated by commas.      |
|                                         |
+-----------------------------------------+
```

The number of subscript expressions must be equal to the number of dimensions in the array declarator for the array except in an EQUIVALENCE statement (see Chapter 7).

A subscript expression is an integer, real or double precision expression. Subscript expressions which are not integer will be truncated to an integer value (see Chapter 11).

An array name may appear without a subscript in:

a. The dummy argument list of a subprogram.
b. The actual argument list of a subprogram reference.
c. An I/O list.
d. A COMMON, EQUIVALENCE, DATA or type statement.
e. A READ, WRITE, PRINT or PUNCH statement as the format designator.
f. A READ or WRITE statement as the file designator.
g. A NAMELIST statement.
h. A DEBUG MONITOR or DEBUG DUMP statement.

Under certain circumstances, the subscript value of a subscript in an array element may fall outside the range declared for the array. The rule is: Any subscript for an array is allowed which references an element of a COMMON block containing the array or an element of an array EQUIVALENCE'd to the array or an element of an array associated with the array through argument passing.

Example:

```
COMMON /X/ A(10),Q,B(19)
COMMON T,C(10)
DIMENSION D(10)
EQUIVALENCE (C(8),D(3))
CALL S(A(5))
```

```
END
SUBROUTINE S(F)
COMMON /X/ R,E(9)
DIMENSION F(99)
RETURN
END
```

Array Name	Allowable Subscripts
A	1 to 30
B	-10 to 19
C	0 to 15
D	-5 to 10
E	0 to 29
F	-3 to 26

The following are examples of valid array elements:

```
B(I)
AMK 599(6)      (interpreted as AMK599(6))
ZIPWITH(3)      (interpreted as ZIPWIT(3) when used
                where an array element may appear)
I5(I5(3))
ARRAY2(2,2,3)
A(R*T)
Q(3.7)          (interpreted as Q(3))
INT(1+9*SIN(I*PI10))
```

The following are examples of invalid array elements:

```
I(I)            (an array name is not an arithmetic expression)
3ARRAY(3)       (invalid array name)
```

The internal handling of arrays is discussed in Chapter 11.

3. In Chapter 7, replace the section title "Array Declarators" by the following:

ARRAY DECLARATORS

An array declarator declares the size and number of dimensions of an array.

The proper format for an array declarator is:

```
+-----------------------------------------+
|                  a(d)                   |
|                                         |
+-----------------------------------------+
|       where a is an array name          |
|     and d is a list of dimension        |
|   declarators separated by commas.      |
|                                         |
+-----------------------------------------+
```

The number of dimensions in the array is the number of dimension declarators in the array declarator. The minimum number of dimensions is one and the maximum is 31.

The proper format for a dimension declarator is:

```
+-----------------------------------------+
|                 d1:d2                   |
|                   or                    |
|                   d2                    |
|                                         |
+-----------------------------------------+
|where d1 is the lower dimension bound    |
| and d2 is the upper dimension bound.    |
|                                         |
+-----------------------------------------+
```

The lower and upper dimension bounds are real or integer constants or simple variables.

The value of a dimension bound may be positive, negative or zero; however, the value of the upper dimension bound must be greater than or equal to the value of the lower dimension bound. If only the upper dimension bound is specified, the value of the lower dimension bound is one.

An array declarator must precede the first use of that array in an executable, DATA or EQUIVALENCE statement.

Each array declarator is either a constant array declarator or a variable array declarator. A constant array declarator is an array declarator in which each of the dimension bounds is a constant. A variable array declarator is an array declarator in which one or more of the dimension bounds are simple variables. Those bounds that are variables are called variable bounds. Variable bounds must be either declared in COMMON statements preceding the array declarator or be dummy arguments.

In addition, each array declarator is either an actual array declarator or a dummy array declarator. An actual array declarator is an array declarator in which the array name is not a dummy argument. An actual array declarator must be a constant array declarator and is permitted in DIMENSION statements, COMMON statements and type statements. A dummy array declarator is an array declarator in which the array name is a dummy argument. A dummy array declarator may be either a constant array declarator or a variable array declarator. A dummy array declarator is permitted in DIMENSION statements and type statements, and may appear only in function or subroutine subprograms.

The size of a dimension is the value d2-d1+1, where d1 and d2 are the dimension bounds for that dimension. The size of an array is equal to the number of elements in the array and is equal to the product of the sizes of the dimensions specified by the array declarator for that array. The size of an array must not exceed 65,535 elements.

Dimension bounds which are not integer are truncated to an integer value before being used to calculate the size of a dimension or the subscript value of a subscript.

Examples:

The array AR declared by

```
DIMENSION AR(3.78:7.21)
```

has 5 elements: AR(3),AR(4),AR(5),AR(6) and AR(7).

The array R declared by

```
DIMENSION R(X,Y)
```

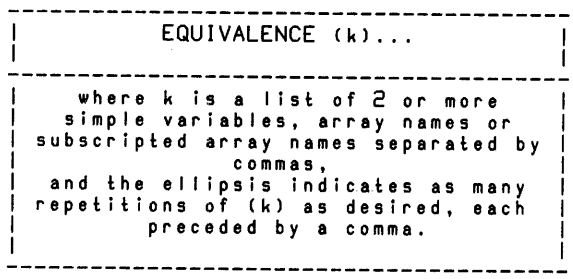
where X=2.8 and Y=3.7 has 6 elements: R(1,1),R(2,1),R(1,2), R(2,2),R(1,3) and R(2,3).

4. In Chapter 7, replace the section on the EQUIVALENCE statement with the following:

EQUIVALENCE STATEMENT

The nonexecutable EQUIVALENCE statement causes two or more variables or arrays referenced in a program unit to share the same memory locations.

The proper format for the EQUIVALENCE statement is:



No dummy argument may appear in an EQUIVALENCE statement. The items appearing in the list k will share some or all of the same storage locations. The subscript expressions of subscripted array names must be integer constants. The number of subscript expressions must be either equal to the number of dimensions for the array or be one. When a subscripted array name in an EQUIVALENCE statement has as many subscript expressions as the array has dimensions, the subscripted array name denotes an array element.

Example:

If the array A is declared by

```
DIMENSION A(-5:5)
```

the statement

```
EQUIVALENCE (A(2),X)
```

equivalences X and the 8th element of array A, which is A(2).

When the name of a multidimensional array appears in an EQUIVALENCE statement subscripted by only one subscript expression, that subscript expression denotes the linearized subscript value of the array element being equivalenced.

Example:

If the array A is declared by

```
DIMENSION A(0:5,-5:-1)
```

the statement

```
EQUIVALENCE (A(2),X)
```

equivalences X and the second element of the linearized form of array A, which is A(1,-5). (See the discussion of how arrays are stored internally, Chapter 11.)

An array name without a subscript denotes the first element of that array.

When two or more subscripted or unsubscripted array names appear in the same list k, the equivalenced arrays are to overlap and be aligned in such a manner that the indicated elements share storage locations. No list may specify more than one element of the same array.

The EQUIVALENCE statement may be used to associate elements with a COMMON block. This association may extend the COMMON block beyond its former terminal point, thus increasing the size of the COMMON block. Such an association may not extend a COMMON block backwards to locations preceding its initial point. Two elements of COMMON storage may not be made equivalent to one another, either directly or indirectly, by an EQUIVALENCE statement.

An EQUIVALENCE statement must precede any executable statement employing items equivalenced.

Storage considerations are discussed in Chapter 11.

Examples:

```
EQUIVALENCE(A,B,C),(A(3),R(9),G)
EQUIVALENCE(D,E)
```

5. In Chapter 11, replace the section title "Arrays" by the following:

ARRAYS

FORTRAN arrays are provided to allow the user to organize program storage locations into a convenient structure. Internally, an array is stored as a group of one or more contiguous data words.

A FORTRAN array of any number of declared dimensions is represented internally by a one-dimensional array of storage locations. Each element of an array has storage requirements and partitioning identical to that of a simple variable of the same type as the array. Thus, each element of an integer array requires one word of storage partitioned like an integer simple variable and each element of a complex array requires two words of storage partitioned like a complex simple variable.

Each integer, real and logical array is allocated as many words of storage as there are elements in the array. Each double precision and complex array is allocated twice as many words of storage as there are elements in the array.

The elements of an array are ordered into a sequence. A subscript of an array element has a subscript value which determines which element of the sequence is identified by the array element. The first element has a subscript value of 1.

Example:

If an array is declared by the array declarator

```
A(L1:U1,L2:U2,L3:U3,...,Ln:Un)
```

the subscript value for array element

```
A(S1,S2,S3,...,Sn)
```

would be

```
1 + (S1-L1)
  + D1*(S2-L2)
  + D1*D2*(S3-L3)
  + ...
  + D1*D2*D3* ... *D(n-1)*(Sn-Ln)
```

where

Dk is the size of the kth dimension, equal to Uk-Lk+1,
where Uk and Lk are truncated to integers
n is the number of dimensions
Sk is the kth subscript expression, which is truncated
to an integer

The offset of an array element within the sequence is its position in the sequence relative to the first element. The offset of an array element is equal to its subscript value minus 1. The offset of the first element of an array is zero.

Subscript expressions are truncated to an integer value before being used to calculate the subscript value.

Example:

For an array declared

```
DIMENSION A(0:5,1:4)
```

the subscript value for A(3.73,2.79) and A(3,2) would be the same:

```
=1+(3-0)+6*(2-1)
=1+3+6
=10
```

The ordering of the array elements produced by the subscript value is identical to that obtained by listing all of the array elements by varying the first subscript expression most rapidly, the second subscript expressions next most rapidly, etc.

Example:

For the array A declared by

```
DIMENSION A(0:2,1:2,-2:-1)
```

the elements are stored in the following order:

```
A(0,1,-2)
A(1,1,-2)
A(2,1,-2)
A(0,2,-2)
A(1,2,-2)
A(2,2,-2)
A(0,1,-1)
A(1,1,-1)
A(2,1,-1)
A(0,2,-1)
A(1,2,-1)
A(2,2,-1)
```

Element A(0,1,-2) corresponds to the first element of the internal array. This order is the order in which array elements are considered when a multidimensional array is used in an EQUIVALENCE statement or when the name of a multidimensional array appears in an I/O list, DATA statement, argument list, COMMON statement, as a format designator, as a file designator, in NAMELIST output, in a DEBUG MONITOR statement or in a DEBUG DUMP statement.

An array is stored internally in consecutive storage locations as long as the storage requirement of the array does not exceed 4095 words. Arrays exceeding this size will be segmented for overlay purposes unless the compiler option LONG is set (see Chapter 19). Each segment will contain a maximum of 256 words and will automatically be overlayed and recalled as necessary during execution of the program. In no case may an array exceed 65,535 elements in length.

6. To Chapter 12, in the section on free-format output, add the following lines after the explanation of the use of the asterisk:

Named output cannot correctly handle subscript expressions whose magnitude contains more than 7 decimal digits. The compiler will give a warning if such subscripts can be detected at compile time. The subscripts printed may be incorrect.

D2403 FORTRAN - WARNINGS ON USE OF "BCL" CONSTRUCTS

Use of BCL constructs will now cause a warning to be emitted indicating that the program is not portable to EBCDIC machines. Only the first occurrence of BCL constructs in a program or (in the case of separately-compiled program units) program unit will cause a warning to be emitted. A warning message will also appear in the trailer information at the end of the listing.

D2446 FORTRAN - "VECTORMODEISALLOWED" DEIMPLEMENTED

The compiler generation option VECTORMODEISALLOWED has been deimplemented. The compiler now has the capability of generating vector mode code by default. It is no longer necessary to compile the FORTRAN compiler with VECTORMODEISALLOWED set to be able to get vector mode code; however, the FORTRAN program must still set the compiler option VECTORMODE to get vector mode code.

D2501 FORTRAN - INSTALLATION INTRINSICS IN "FORTRAN"

It is now possible to write installation intrinsics in FORTRAN.

The compiler option INTRINSICS has been implemented. This option must be set before the first source statement of the compile deck and cannot be set in batch mode. Setting INTRINSICS sets the option SEPARATE, and the subroutines or functions being compiled can be bound into the intrinsics file after compilation. The option LIBRARY may be set explicitly if desired.

An intrinsic must be a subroutine or function subprogram. The compiler options OWN, OWNARRAYS and LEVEL cannot be set in an intrinsic. The following items cannot appear in an intrinsic:

a. a STATISTICS or DUMP STATISTICS statement
b. a STOP or CALL EXIT statement
c. a DATA statement or an initial value list
d. a COMMON statement
e. an EXTERNAL statement
f. an ENTRY statement
g. formal subprograms
h. file declarations
i. a PRINT or PUNCH statement
j. a CHANGE, OPEN, INQUIRE, LOCK, PURGE, CLOSE, FIND, BACKSPACE, ENDFILE or REWIND statement
k. a READ or WRITE statement not specifying core-to-core I/O
l. a free-format designator containing an asterisk (*)

D2532 FORTRAN - COMPILER FILE LINE

Table 18-1 of the FORTRAN Manual (Form 5000458) should be corrected. The COMMENTS for file LINE should say:

> "Optional and label-equatable to REMOTE file produced when either the compiler option LIST or TIME is set."

D2573 FORTRAN - EXPONENTIATION MEANING

The following should be added to the FORTRAN Reference Manual, Form No. 5001506, after page 6-2:

The meaning of exponentiation (**) depends on the values of the operands. The various meanings of Y**Z are as follows:

	Z-TYPE			Z-TYPE		
	INTEGER			REAL or DOUBLE		
Value	Z>0	Z=0	Z<0	Z>0	Z=0	Z<0
Y>0	Note 1	1	Note 2	Note 3	1	Note 3
Y<0	Note 1	1	Note 2	Note 4	1	Note 4
Y=0	0	Note 4	Note 4	0	Note 4	Note 4

Note 1: Y**Z=Y*Y*Y...*Y (Z times)

Note 2: Y**Z=the reciprocal of Y*Y*Y..*Y(ABS(Z)times)

Note 3: Y**Z=EXP(Z*ALOG(Y))

Note 4: Value of expression is undefined

FORTRAN

P1365 FORTRAN - "INVALID OP" IN "CHANGE" , "OPEN" , "INQUIRE"

An INVALID OP error, which occurred at run time when a formal array was used to contain or receive a pointer-valued attribute such as TITLE in a CHANGE, OPEN or INQUIRE statement, has been corrected.

P1367 FORTRAN - CORRECTION TO "VECTORMODEISALLOWED" REMOVAL

A syntax error no longer occurs if the FORTRAN compiler is compiled with the compiler generation option OPTLSS1 set.

P1494 FORTRAN - FORMAL ARRAY IN IMPLIED "DO" LOOP

An INVALID INDEX would occur when an attempt was made to initialize a formal parameter using an implied DO loop in a DATA statement if the formal parameter was not dimensioned.

Formal parameters cannot be initialized in DATA statements; this error will now be flagged correctly.

P1506 FORTRAN - UNEXPECTED END OF INITIAL VALUE LIST

The compiler would get an INVALID INDEX when the end of an initial value list occurred unexpectedly; e.g., REAL A(10)/1,2,3. This has been corrected.

P1541 FORTRAN - FORMALS AS "IMPLIED-DO-VARIABLE"

When a formal parameter was used as an implied-DO-variable in a DATA statement, the compiler would get into an unending loop. This problem has been corrected. The use of a formal parameter in this manner will not affect the value of the parameter during execution.

P1542 FORTRAN - SOURCE PROGRAMS IN "BCD" CODE

FORTRAN source programs with dollar option BCD set were being interpreted as BCL instead of BCD. This problem has been corrected.

P1543 FORTRAN - SYNTAX FOR "DO" STATEMENT

Certain incorrect delimiters were being accepted in place of a comma between the initial and terminal parameters in the DO statement. The correct syntax error will now be emitted.

P9068 FORTRAN - "OPT=1" READLOCK CODE

The optimizing compiler would produce incorrect code for sequences of statements such as the following, where "R" is an array:

```
TB=2.*R(II)
R(II)=R(I)
R(I)=TB
```

The code produced would in effect perform the last assignment before the second-to-last, causing incorrect results when I NEQ II. The optimizing compiler has been corrected to emit the correct code in these situations.

P9292 FORTRAN - "$" OPTION "LEVEL=N," N>15

An attempt to set the lex level of a program or program unit to a value greater than 15 through the use of the LEVEL compiler option will result in a warning message "LEX LEVELS > 15 WILL BE DEIMPLEMENTED".

P9293 FORTRAN - "$INCLUDE" OF A NULL RANGE

If a $INCLUDE card with an internal file name specified a line range which contained no records, a subsequent $INCLUDE card on the same internal file name specifying a line range preceding the original range would cause no records to be included. This problem has been corrected; now, records within the second line range will be included.

P9294 FORTRAN - IMPROVEMENTS TO TRAILER INFORMATION

Improvements have been made to the trailer information printed at the end of a listing. The last line of trailer information has been broken up into two lines, the first line giving the code file title and the second line giving information about the compiler.

P9295 FORTRAN - "INVALID INDEX" ON OVERSIZED COMMON BLOCK

An INVALID INDEX which would occur in the compiler under certain circumstances in OPT=0 when a common block was more than 65535 elements long has been corrected. The following example exhibited the problem:

```
COMMON C(11),Q(432000),OT(2700)
EQUIVALENCE (C(11),TAUH)
DATA TAUH/3./
END
```

P9296 FORTRAN - USE OF "EXIT" AS A SUBROUTINE

The OPT=0 compiler was treating a call on a subroutine named EXIT with parameters which had not yet been compiled as the CALL EXIT statement. Now, such a statement will be treated as a call on a subroutine named EXIT.

INPUT-OUTPUT

D2287 IN-OUTPUT - "I/O" SUBSYSTEM IMPROVEMENTS

Improvements have been made to the I/O Subsystem:

1. File Open errors now show a character string describing the error instead of returning just an error number.
2. An FA response to a File Requires message that changes the KIND to DISK, PACK or REMOTE now will work instead of remaining in a File Requires <new kind> state.

D2498 IN-OUTPUT - BACKUP FILE FEATURES

Four features have been added to the format of backup files. Each change is upward compatible and will not affect programs reading backup files in a correct manner. Autoprint and SYSTEM/BACKUP have not been changed.

The features are as follows:

1. Word [1].[3:1] of the header record entry will contain a one if the file is a backup disk file, the labeltype of the file is STANDARD and the label entries are not present (this can occur when the PRINTERLABELS SPO option is reset).
2. The usercode, chargecode and task name in the header record entry will no longer be truncated.
3. Word 298 of each block will contain the number of records in each block. Word 299 of each block will contain the record number of the first record in each block.

   Printer/punch label entries and the header record entry are not included in the record counts. Each space 2 entry counts as 2 records. All other entries count as one record.
4. Word 9 of the header record is now defined to hold the level number of the backup file. All backup files created prior to these additions will have a zero in this word. This word will now contain the current backup level of one to indicate the presence of these additions.

D2516 IN-OUTPUT - "BCL" BACKUP FILES

As an aid in making backup files portable to EBCDIC machines, all BCL backup files will now be translated by logical I/O's software translation to EBCDIC as they are being created. The change should be transparent when the files ultimately are punched or printed.

On III.0, printer or punch backup files that explicitly set EXTMODE=BCL will be excepted from the conversion to EBCDIC.

On III.1, only those printer or punch backup files that explicitly set EXTMODE=BCL and go to a backup 7-track tape will be excepted from the conversion.

On III.2, there will be no exceptions.

INPUT-OUTPUT

P1325 IN-OUTPUT - "FILETYPE=6 RECORDCOUNT"

The FILETYPE equal to six logical I/O intrinsics incremented the RECORDCOUNT before finding the record instead of after. This caused the input reelswitch logic to see a record count mismatch when reelswitching tape files. This has been corrected.

P1335 IN-OUTPUT - "FILETYPE=4" RANDOM "I/O"

Random I/O to a FILETYPE equal to four file would fail with data errors if the file was character mode and the attribute SIZEMODE equaled SINGLE. This has been corrected.

P1382 IN-OUTPUT - "CARRIAGECONTROL=CTLASA" OR "CTL360"

The PL/I and FORTRAN character-oriented printer files have their carriage control embedded in the first character of each record. When records in this format are used with the <carriage control or record number> syntax of ALGOL or COBOL (e.g., WRITE(F[SKIP 1],132.A);), the <carriage control or record number> form of control is ignored and the information embedded in the record is used instead.

Previously, if a write statement were used for control only (e.g., WRITE(F); or WRITE(F[SPACE3]);), the system would get a SEG ARRAY interrupt fault in the I/O routines. This has been corrected.

P1410 IN-OUTPUT - STATUS CHANGE VS. "RSVP" REPLIES

The system will now be more responsive to operator replies to NO FILE system output RSVP messages.

P1574 IN-OUTPUT - "LEB" RESIZE MISSED BY "OUTPUTREELSWITCH"

The process of switching reels of an output tape file could get a fault in DO code if the operator responded to the FILE REQUIRES RSVP with an FA response. This problem is now eliminated.

P9273 IN-OUTPUT - "PROGRAMDUMP," NEW "IOCB"

The output of the file section of PROGRAMDUMP has been modified to correctly identify the new words in the IOCB.

Tape labels now contain four-digit system serial numbers. The USYSID field in the HDR2.EOF2 label now begins at character number 45 (starting at character one). The old three-digit field, beginning at character 37, will contain the lower three digits of the system serial number until the III.1 release, at which time the three-digit field will change to blanks. The tape level value (characters 35 and 36 of the VOL1 label) have been changed to 3.

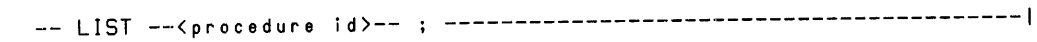
P9274 IN-OUTPUT - "CLOSE HERE" ON EMPTY TAPE FILE

A COBOL CLOSE HERE statement executed upon an empty tape file will no longer either cause a block count error or an incorrect number of records being written on the tape.

INTERACTIVEXREF

D2425 IXREF - "XREF" PROCEDURE END SEQUENCE NUMBERS

The Interactive Xref program will now list the ending sequence number as well as the starting sequence number for ALGOL and ESPOL procedures wherever it had listed the starting sequence number previously. A procedure may also be listed through Interactive Xref by using the following syntax:



D2525 IXREF - ADD "PRINTER" OPTION FOR "HELP" COMMAND

The HELP command now recognizes PRINTER as a valid option. The PRINTER option functions in the same manner as it does when used with other commands.

Example:

```
HELP:PRINTER
```

D2526 IXREF - CHANGE "INTERACTIVEXREF" FILE TITLES

The format of the titles of the files generated for INTERACTIVEXREF has been changed to the following:

```
<usercode>XREFFILES/<codefile title>/DECS
<usercode>XREFFILES/<codefile title>/REFS
```

D2527 IXREF - VERSIONS FOR "INTERACTIVEXREF" FILES

Version information is now included in the XREFFILES. INTERACTIVEXREF checks the compatibility of the XREFFILES and displays an appropriate error message if the XREFFILES were created with an incompatible XREFANALYZER.

INTERACTIVEXREF

P1430 IXREF - MODIFY OUTPUT FORMAT

The characteristics of certain screen terminals will no longer cause the output format used by INTERACTIVEXREF to create a subsequent user problem.

P1431 IXREF - CORRECTLY EXPAND DEFINES

Define text will now be correctly recognized if it follows a quoted string and is on the same line.

JOB FORMATTER

D2486 JOBFORMAT - "PE9MT" VS. "PETAPE"

PE9MT in JOBFORMATTER's UNITMNEMONICS array has been changed to PETAPE to make it consistent with other software. This change will affect JOBFORMATTER output, LOGGER and LOGANALYZER.

JOB FORMATTER

P1293 JOBFORMAT - SWAPJOB NUMBER AND JOB NAME

The position of job name and job number have been exchanged on the banner page of WFL output.

P1294 JOBFORMAT - COSMETIC CHANGES

Several cosmetic flaws in LOGANALYZER output for IOERRORs and MAINT requests have been corrected.

P1422 JOBFORMAT - CHECKBIT RESULT MAPPING

The checkbit field of interrupt parameter P3 on the B6800 is now displayed after mapping into a bit number.

P1475 JOBFORMAT - JOB LOG "I/O" ERRORS

JOBFORMATTER will now stop printing the job file when an I/O error occurs while reading from the job log.

LCOBOL

D2417 LCOBOL - "FILEKIND" FOR "LCOBOL" SYMBOL, CODE FILES

If a new LCOBOL source file is created using the $NEWD option, the FILEKIND will equal "LCOBOLSYMBOL". The codefile created by the compiler will have FILEKIND equal "LCOBOLSL3CODE" or "LCOBOLSL5CODE", depending on whether SL3 or SL5 is desired.

LCOBOL

P1368 LCOBOL - CORRECT BAD LITERALS, "OPTIM"

The LCOBOL compiler will now process literals correctly.

The OPTIMIZE dollar option now works correctly.

P1593 LCOBOL - PATCHING "LCOBOL" SOURCE PROGRAMS

The LCOBOL compiler was not patching some programs correctly. This has been corrected.

P1608 LCOBOL - NUMBER OF CHARACTERS IN A GROUP MOVE

In some cases the number of characters moved for a group move was incorrect. This has been corrected.

LOADER

P1343 LOADER - CHANGE PANEL, "SPO" DISPLAY

The processor panel display has been changed to show the correct LOADER level. The SPO BOJ message has been changed to indicate the appropriate machine (e.g., B6700, B6800).

P1455 LOADER - DISK PACK "I/O" ERRORS

The LOADER will now recognize I/O errors on disk packs/

P1460 LOADER - CORRECT "EOJ" MESSAGE

The EOJ message will now indicate B6800/B6700 as appropriate.

P1551 LOADER - "LOADER" VS. MULTIPROCESSOR

The problem that was causing a multiprocessor system to superhalt after issuing the LOADER EOJ message has been corrected.

LOG ANALYZER

D2554 LOGANALY - "B6800 CPUERROR"

A new option CPUERRORS, which applies only to the B6800, has been added to LOGANALYZER. It produces a chronologically analyzed list of all system error interrupts. The following are the interrupts that are logged:

Alarm
General control
Hardware

For a complete description of the interrupts, refer to the B6800 System Reference Manual (Form No. 5001290), Section 5 - System Concept.

Note: The error interrupts will also appear for a "LOG MAINT" request.

D2561 LOGANALY - "PROCESSOR ID"

LOGANALYZER will now display the "PROCESSOR ID" for all B6800 ERROR INTERRUPTS for the "CPUERRORS" and "MAINT" log requests.

LOG ANALYZER

P1133 LOGANALY - "SO" AND "RO" OPTIONS DISPLAY

LOGANALYZER will now display a list of SO or RO option numbers separated by commas.

P1294 LOGANALY - COSMETIC CHANGES

Several cosmetic flaws in LOGANALYZER output for IOERRORs and MAINT requests have been corrected.

P1295 LOGANALY - "STOPDATE" DEFAULT

The STOPDATE default for LOGANALYZER has been changed from 99359 to 99367 to correct a possible error at the end of any year.

P1296 LOGANALY - "MAINT" LOG ENTRIES

LOGANALYZER will now sort MAINT log entries by date and then by time; previously, they were sorted by just time.

P1297 LOGANALY - "IOERRIRRCOUF" FIELD

LOGANALYZER will no longer try to store the number 4 in the IOERRIRRCOUF field, which is only a 2-bit field.

P1298 LOGANALY - "INVALID INDEX" IN "SORTIN" PROCEDURE

LOGANALYZER will no longer get an INVALID INDEX in the SORTIN procedure by not properly resizing the JOBSWITHERRORS array.

P1573 LOGANALY - "132" CHARACTER OUTPUT

LOGANALYZER will now use an output buffer size of 132 characters for its print files.

P1580 LOGANALY - "LOG COMMENT" EQUALS "LOG OPERATOR LC"

LOGANALYZER will now handle "LOG COMMENT" and "LOG OPERATOR LC" in the same way as log entries made with the ODT requests "LC" and "LJ" are handled.

P1637 LOGANALY - "LOG MAINT" CORRECTION

The headings for LOG MAINT displays has been corrected to match the changes made in the data displayed. The MPX and RESULTDESC, etc., headings will now line up with data displayed below them.

P9026 LOGANALY - SYNTAX CORRECTION FOR "SESSION"

LOGANALYZER will now correctly syntax the word "SESSION" and properly handle the "AP" SETSTATUS request.

P9032 LOGANALY - "UA/UR" MESSAGE DISPLAYS

LOGANALYZER will now display the MPX and PATH information for UR's and UA's involving disk packs.

P9052 LOGANALY - SPURIOUS RECORDS IF FILE EQUATED TO DISK

LOGANALYZER will now allow the LINE file to be file-equated to disk and produce the same output as on the printer. The problem was in the printer output that was doing SPACES which appeared as garbage records on disk.

P9203 LOGANALY - LOG DATE CORRECTION

LOGANALYZER will now display the correct time and date range for any released SUMLOG file and the current SYSTEM/SUMLOG file when the DATE option is used.

P9302 LOGANALY - MAINTENANCE LOG CHANGE

Mark II.8 note P5579 should be changed as follows:

Log entries are no longer grouped by entry type, but are listed in time order within jobs.

LOGGER

D2536 LOGGER - "JOBSUMMARY" FILE RECORD SIZE INCREASE

The JOBSUMMARY file record size in LOGGER has been increased from 60 to 90 words to allow more information to be maintained. LOGGER will be able to read the old 60-word record files, but all new output JOBSUMMARY files will be 90 words. Any field that is used for output will be zero if it falls beyond 60 words when processing old size JOBSUMMARY files.

D2537 LOGGER - "PRE27" OPTION REMOVED

The run-time option "PRE27" has been removed from LOGGER.

D2538 LOGGER - "JOBENTRYTIME" , "JOBQUEUEDTIME"

LOGGER will now display the JOBENTRYTIME of a job from its BOJ log entry. This item is stored in the JOBSUMMARY file and is type STRING. The JOBQUEUEDTIME is also stored in the JOBSUMMARY file and is of type real. JOBQUEUEDTIME is the difference between JOBENTRYTIME and STARTIME in minutes. These two items only have meaning for jobs; they do not apply to tasks.

Missing log records (unmatched BOJ and EOJ) will cause errors in these two items.

LOGGER

P1299 LOGGER - "INCLUDE, EXCLUDE" CORRECTION

LOGGER will now handle the following correctly; i.e., it now works like the II.8 release:

```
INCLUDE <text> NEQ " "
EXCLUDE <text> NEQ " "
```

P1300 LOGGER - THREE VS. FOUR DIGIT SYSTEM "ID"

LOGGER will now handle any length (1 to 17 digits) system identifiers when looking for SUMLOG files.

P1642 LOGGER - VERSION HEADING CHANGED

LOGGER will now display its complete version in its heading line.

P1643 LOGGER - LOG TITLE DISPLAY TRUNCATED

A possible error in LOGGER has been corrected that could cause a log title to be truncated at 21 characters.

P1644 LOGGER - "REAL, INTEGER ITEMS" USED AS "BREAK ITEMS"

LOGGER will no longer get INVALID INDEX faults when ITEMS of type REAL or INTEGER are used as BREAK ITEMS.

Example:

```
BREAK ON PRIORITY
BREAK ON ELAPSEDTIME
```

P1645 LOGGER - "PRIORITY" ITEM TYPE "INTEGER"

The PRIORITY item type has been changed to INTEGER to conform to the System Software Operational Guide, Volume 2, No. 5001688.

P1646 LOGGER - "INCLUDE, EXCLUDE" VS. STRINGS

LOGGER will now make all INCLUDE and EXCLUDE comparisons using the minimum length of the two string. This corrects the problem where strings had to be padded with sufficient blanks to meet the length as specified in the appropriate IDTABLE.

P8618 LOGGER - "UPDATE" OPTION NEW FILE CREATION

The UPDATE option will now copy all the records in the "old" YTDFILE to the "new" YTDFILE when LOGGER is run twice in the same day.

The data in a YTDFILE will not be lost when an UPDATE is requested that crosses a new year. The task will stop with a date comparison error and not purge the old YTDFILE.

P9118 LOGGER - "INCLUDE,EXCLUDE,CORRECTION" ARRAY RESIZE

LOGGER will now allow as input as many INCLUDE, EXCLUDE and CORRECTION cards as necessary and not die with INVALID INDEX. The array used by these input types is now RESIZED at run time as needed.

P9119 LOGGER - "EOF NO LABEL" IF INPUT EXCEEDS "60" CARDS

LOGGER will now allow input decks of more than 60 cards.

P9204 LOGGER - RETENTION REPORT ITEM CORRECTION

The RETENTION report item for FILEIODATA in LOGGER will now be displayed for disk packs as well as disk units.

P9205 LOGGER - INCLUDE "UNITNO" CORRECTION

The FILEIODATA item UNITNO is now a numeric item type and not a string. This allows UNITNO to work correctly when used in an INCLUDE or EXCLUDE statement.

P9206 LOGGER - "BREAK ON PRIORITY"

LOGGER will no longer get an INVALID INDEX when a control BREAK ON PRIORITY is requested.

LTTABLEGEN

P1124 LTTABLEGEN - USER SPECIFIED TABLES

SYSTEM/TRAINTABLES will now be properly modified to reflect user specified train tables.

P1130 LTTABLEGEN - "450/750 LPM" TRAIN PRINTERS

450/750 LPM train printers will now have appropriate train tables loaded for the following trains:

16 char	EBCDIC-3	-TRAINID=1
64 char	EBCDIC-3	-TRAINID=5
64 char	SWEDEN OCR-B	-TRAINID=14

Note: Although the appropriate tables are loaded for TRAINID's 1 and 5, they will show respectively as EBCDIC18 and EBCDIC72 instead of EBCDIC16 and EBCDIC64. This discrepancy will be resolved later.

MAKEUSER

D2282 MAKEUSER - USE OF "USERDATALOCATER"

If USERDATALOCATOR's parameter is a literal string instead of a pointer, the intrinsic will be invoked at compile time. If the resulting value indicates that an error occurred, the compiler will give a warning and generate a run-time call. If the resulting value does not indicate an error, that value will be used by the compiler as a constant.

MCP

D2246 MCP - "B6800 MCP"

New hardware interrupt literals are now recognized, and, where appropriate, logged.

The format of log records reporting B6800 processor errors or diagnostic interrupts is as follows:

Word	Contents	
0	Link	
1	Date	
2	Time	
3	Type	31:16 = 2 (LOGMAJMAINT) 15:16 = 15 (MLCONFIDENCEERROR)
4	MCPID	47:12 = System Serial # 35:12 = Mark Digit 23:12 = Mark Level 11:12 = Patch Level
5	ProcessorID, WHOI Value	
6	P1 Parameter, P1 Interrupt Parameter with Tag set to 0	
7	P3 Parameter, P3 Interrupt Parameter with Tag set to 0	
8	P2 Parameter Tag	
9	P2 Parameter, P2 Interrupt Parameter with Tag set to 0.	

The interpretation of P1, P2 and P3 is based on B6800 processor specifications.

D2249 MCP - "OF" OF "GETSTATUS" WAITING ON <FILENAME>

OF has been added to the possible replies of the MCP GETSTATUS RSVP of "WAITING ON <filename>". The possible replies are now "OK", "DS" and "OF".

D2268 MCP - IGNORE UNKNOWN UNIT TYPES

The MCP will now ignore units which are present at Halt/Load time but have a unit type (as supplied by the PC encoder card) which is greater than any type the MCP is programmed to handle. Previously, presence of such a unit on the system would cause Halt/Load initialization to fail with an INVALID INDEX.

Note that it is still not possible to Halt/Load with an unknown unit that uses a status-change interrupt, since the MCP must correctly move these by control and unit. For example, this change makes it possible for an older MCP to ignore floppy disk but not model 235 diskpacks.

D2269 MCP - "RESERVE" ENHANCEMENTS

The operational characteristics of RESERVEDISK have been improved. Previously, RESERVEDISK would hang waiting for exclusive use of a file before attempting to copy it. This caused delays - some of quite extended duration - during the pass of the permanent files residing in the area being reserved. RESERVEDISK may now continue through the list, and later recycle through it on operator request, presumably after some action has been taken to correct the blockage. If the operator is unable to correct the hangup, he may "QT" RESERVEDISK, which will create a new directory of XDISK titled "RESDISK".

RESDISK is an XDISK file which is titled in the same way as "BADDISK":

RESDISK/FMLYINX<family index no.>/UNIT<unitno>/AD<address>.

If a RESDISK file is encountered on a subsequent reserve, it will be removed. The effect of this is that any part of the old RESDISK which intersects with the new RESERVEDISK specification is absorbed into the new set of reserved disk, and the non-intersecting part is released as available for system use. The rationale of this operation is that it is a simple solution to an otherwise complex naming problem.

RESERVEDISK, at the end of a pass through the permanent files which intersect the area being reserved, will inform the operator of any files which could not be moved. The operator may then take one of three actions, as follows:

DS Releases all space reserved during this run of RESERVEDISK. This is equivalent to II.7 action.

OK Causes another pass through the permanent intersecting files. If any are still blocking the reserve, they will be displayed. If no blockage now exists, RESERVEDISK will proceed to handle temporary files.

QT Causes creation of RESDISK files out of areas which would otherwise be released, and then send the reserve to end-of-job. This option may be used to cut off the RESERVEDISK overhead, while preventing the subsequent allocation to areas which later may be required by a reserve.

RESERVEDISK will move executable and intrinsic files (except JOBCODE) even if they are open. Access to the headers is obtained with a "conditional exclusive" request and is gained if the file is not being used; otherwise, the file is opened as "shared". This may prove valuable for files such as CANDE and intrinsics, which are accessed via PRESENCEBIT and not logical I/O; therefore, disk addresses will not be left floating in buffers. Since they are read-only files, no WRITE to the wrong area after copying will occur. If a datafile is encountered open with only inactive rows intersecting the reserve area, these rows will be moved up to the first active row encountered.

The job number of the first stack to open a temporary file is displayed, if the stack is still alive, which should assist the operator in identifying which job is blocking the reserve. At this time (during a hang on scanning for the existence of temporary files), the operator may do a QT, the effects of which are described above.

Old BADDISK files are retained in the directory, even though the rows are released, if a subsequent RESERVEDISK absorbs them. The rowsize is set to zero, identifying the situation to the operator on a "PD" message.

If a BADDISK file exists and a reserve is done overlapping its high order address end, RESERVEDISK will no longer hang on a duplicate name.

D2273 MCP - "IV" VS "235"

The IV routines for 235 diskpacks have been changed. 235 packs come from the factory already initialized. If the packs are damaged, they must be returned to the factory to be reinitialized.

The INITIALIZE and VERIFY routines in the MCP or LOADER must be used to put a label on 235 packs. However, these routines will not reinitialize any cylinders on the 235 packs.

D2439 MCP - DIRECTORY CONVERSION

The MCP will no longer recognize or convert disks or packs that use pre-II.7 directory format.

D2440 MCP - "NEWFILE" ATTRIBUTE

A new file attribute has been implemented to specify whether a new file is to be created or an existing file is to be accessed.

The attribute name is NEWFILE; its number is 136. Its usage and access specifications are:

    General, read/write, anytime/closed, Boolean

The attribute actually has three states: never-set, TRUE and FALSE. If the attribute has never been set, then it has no effect: a new DISK or PACK file will be created if the AREASIZE attribute has been set and the MYUSE attribute is explicitly or by default OUT. If NEWFILE is set TRUE, then a new file will be created irrespective of the settings of other attributes. In the case of DISK and PACK files, a default AREASIZE will be chosen by the MCP if none was specified. If NEWFILE is set FALSE, then an existing file will be sought, irrespective of the settings of other attributes. A file-open error results if NEWFILE is set incompatibly with the device type, such as FALSE for a PRINTER or TRUE for a READER.

The NEWFILE attribute may be read at any time: the value is TRUE if NEWFILE has been explicitly set TRUE and FALSE otherwise. If the attribute has not been set, the value is FALSE whether or not a new file was created according to the default criteria.

D2447 MCP - "COPY" VS. SERIAL NUMBERS

Library Maintenance has been changed to use serial numbers when they are specified for source disk and destination disk volumes.

Note that when a non-zero serial number is specified, family substitution does not take place.

D2496 MCP - SWAPPER ENHANCEMENTS

The III.0 release contains several enhancements in the area of SWAPPER and swapped jobs. These include the following:

SWAPPER Parameters:

The SWAPPER parameter CORESIZE may now be changed via the AX input. If it is decreased, the core in use by SWAPPER will be reduced as soon as tasks occupying the area to be returned to the system have been swapped out. If CORESIZE is increased, the new value will be recorded in SYSTEM/SWAPDISK and used the next time that swapper is initiated.

SWAPPER I/O Errors:

If an irrecoverable error occurs during a write of a swap task to disk, a new disk area is allocated for the task and the write is reinitiated. If an error occurs during the read of a swap task, the task is error terminated and an attempt is made to return the resources in use by that task.

Any area of SYSTEM/SWAPDISK that is involved in an irrecoverable error will be recorded in the swapdisk preamble and that space will not be reused. All such discarded areas are displayed in response to an AX BADDISK input to swapper.

Operator Inputs:

The <mix number> and <mix number>OT inputs can be used for swap tasks.

D2518 MCP - SEGMENTED STRING VARIABLES

An ALGOL string is initialized to a maximum length of 132. If this length is exceeded, the string is resized (to multiples of 32 words) to a maximum of 256 words. If this length is exceeded, the string is segmented and increased in increments of 256 words to its specified maximum size (currently 65534 characters).

D2539 MCP - "OPTIMIZER" OPTION REMOVED

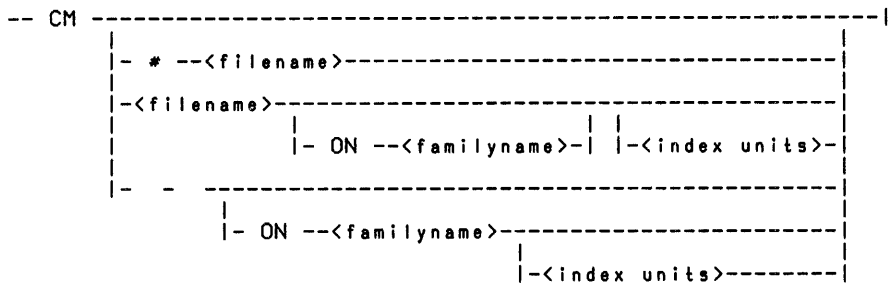
The MCP compile-time option OPTIMIZER has beem removed from the MCP and Maintenance symbolics.

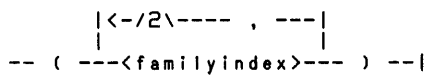
D2550 MCP - RETURN QUEUE OF ORIGIN

In Case 3 or Case 9 SYSTEMSTATUS requests, the job class information will be put in the array passed to SYSTEMSTATUS after the task or compilee name. If bit 46 of the word returned is not on, there is no queue of origin. If bit 46 is on, the queue of origin is in bits 19:10.

D2569 MCP - "CM" COMMAND

The ODT Reference Manual, Form No. 5001704, Page 2-19, has been corrected. Complete new syntax and semantics for the CM command follow.





The CM (Change MCP) message changes the running MCP to the specified MCP.

Examples:

```
CM
--
```

The name of the next MCP, if any, is displayed.

```
    SYSTEM/UPDATE/MCP WILL BE THE NEXT MCP
```

```
CM # SYSTEM/X
-------------
```

The MCP specified will be loaded into memory when the mix count is zero. The bootstrap records are not updated. A Halt/Load causes a reversion to the original MCP.

```
CM SYSTEM/UPDATE/MCP
--------------------
```

The MCP waits for a null mix, finds the specified file on the current Halt/Load family, makes a copy of the MCP on each family index that currently has a copy of the running MCP, and simulates a Halt/Load from the new MCP. The name of the next MCP is displayed, as follows:

```
    SYSTEM/UPDATE/MCP WILL BE THE NEXT MCP
```

Then

9734 *** CM IN PROGRESS ***

CM IN PROGRESS will be displayed for each row of the MCP code file that must be moved to the Halt/Load unit.

CM SYSTEM/X ON PACK

The CM message can specify an MCP code file on other than the Halt/Load family by using the ON <familyname> syntax, where the familyname is a disk or pack familyname. In this case, CHANGEMCP is simply started as a visible independent runner without waiting for the mix count to go to zero. The particular disk or pack is set up as an MCP system disk that can be used for Halt/Loading. In order to actually run the new MCP, it is necessary to Halt/Load from the disk after CHANGEMCP goes to end-of-job.

CM SYSTEM/X ON PACK (IND1,IND2)

Two copies of the MCP code file are created: one on each of the familyindex units, IND1 and IND2. All units receiving MCPs must have a directory present at CM time; otherwise, the CM procedure will say the following:

"CM INVALID FAMILY INDEX OR MISSING DISK"

(See the DIRECTORY DUPLICATE (DD) command.)

All MCPs will be given a suffix of the form FMLYINX<nnn>, where <nnn> is the familyindex number.

When using multiple MCPs, MCP presence bit I/Os are rotated among the list of MCPs.

To get the default MCP configuration, use the following:

"CM SYSTEM/MCP (1)"

CM -

An MCP change waiting for a mix count of zero is cancelled.

CM - ON PACK

CM - ON PACK (IND1,IND2)

The bootstrap is erased from the indicated Halt/Load family and the MCP code file may be removed by the operator.

D2577 MCP - CLOSE "REEL" STATEMENTS

A request for a reel switch (e.g., a COBOL CLOSE REEL statement) will no longer leave an open file in the closed state if the file's kind is not a magnetic tape device. Instead, no action will be taken at all, leaving the file in the same logical state (i.e., open) that it would be if its device kind were magnetic tape.

A request for a reel switch on a closed file will continue to attempt to do a normal close on the file.

D2579 MCP - "PRE-2.4 ON <FAULT> STATEMENTS"

The MCP version 3.1 will no longer support the 'ON <fault>' statements compiled by the ALGOL, DCALGOL or ESPOL compilers prior to the 2.4 release. An entirely new mechanism was implemented in 2.4 (released mid-1973).

Any existing code file which was compiled with a 2.3 or older compiler and which uses 'ON <fault>' statements must be recompiled prior to being run on a 3.1 system.

If such a code file is run on a 3.0 MCP, the following warning message will be generated the first time an 'ON' statement is executed:

BEFORE 3.1, RECOMPILE PRE-2.4 CODE WITH 'ON' STATEMENT @ <addr>

The <addr> is the line number, or seg:word:syl address if no lineinfo is present, for the 'ON' statement. Only one message is generated for any one task. The message will appear in the job log, and will be sent to the user's terminal if the task was run via CANDE and the user's MESSAGE options is set.

Note that the 'ON' statement is "executed" whenever it appears in the flow of control, to arm the fault trap. It is at arming time, not fault-detection time, that the program will be aborted.

MCP

P1126 MCP - STICKY MEMORY

The sticky memory function available to DCALGOL via the resident function may now be used in the MCP by calling the procedure MAKERESIDENT.

P1182 MCP - "DBS D3 NOMEM"

DMSOPEN will now hang with an RSVP if insufficient memory is available for the D3 stack. If this RSVP is DS'ed, DMSOPEN will return ERROR 42.

P1224 MCP - AVOID "GO TO" EXITED BLOCK

An attempt to perform a GO TO from a procedure invoked by an EPILOG to a label declared in the same block as the EPILOG will be intercepted and handled as though the label had been at the very end of the block. No further statements will be executed in that block (which is being purged at block-exit when the EPILOG was invoked).

P1233 MCP - SPLIT "DBS" INTO ACTIVE/NONACTIVE

The DBS bit in stackinfo is now split into two bits. One bit identifies the stack as a DBS; the other bit will be true if and only if the DBS is active. This corrects a problem which resulted in nonactive DBS stacks being left around due to the use of the "visible DBS".

P1313 MCP - "LINKLLISTIT"

FLATREADER no longer calls LINKLISTINSERT so often. When doing catalog rebuilds, etc. it no longer processes irrelevent directory rows. Consequently, less processor time is used.

P1314 MCP - "CP" VS. "HPT/PACK"

When a job that takes a checkpoint to a PACK(DISK) goes to EOJ, it will not try to REMOVE (or CHANGE) the CP file on DISK(PACK) if no CP file was put there.

P1316 MCP - "CP" FILE SECURITY

Checkpoint files will now have CLASSA IO security.

P1317 MCP - LOCKS VS. "PAST"

Doing a VOLUME DELETE and powering down unvolumed disk units will no longer hold up normal directory accessing of other disk units.

P1323 MCP - FILE SIZE VS. CLOSE WITH CRUNCH

The MCP will now log the correct FILE SIZE when a file is closed with CRUNCH.

P1324 MCP - MAKE "WFL" USE DIRECT ARRAY

The interface between the operating system and WFL has been changed to use a direct array parameter.

P1326 MCP - "INFODUMPER" OPTION

A user option, INFODUMPER, has been defined to facilitate creating info files. If INFODUMPER is set, DUMPINFO will be performed at 18999997.

P1327 MCP - ERROR MESSAGES FOR UNRELEASED ATTRIBUTES

If a program attempts to use an unreleased task attribute or file attribute, it will now be DS'ed with an appropriate message.

P1328 MCP - FOREIGN PRINTER TRAIN PROBLEM

A problem has been corrected which affected a foreign printer train. This correction does not affect domestic line printers.

P1329 MCP - SPECIAL PERIPHERAL CONTROL

A new unit type has been implemented.

P1330 MCP - "GETSTATUS" VS. DIRECTORY

LISTVOLUMELIB will no longer allocate a disk file as big as the system catalog.

FLATREADER errors will now cause dumps if the MCPTEST option is set.

PD to a off-line volumed disk will work again.

P1331 MCP - "READER BADMOM"

The LA array, which was being incorrectly reused when looping to read from a backup directory, has been removed.

P1333 MCP - "PATHRES PBIT"

The following function has been corrected:

UA PK<nn> MPX<m> PATH<p>

P1334 MCP - "CM" SUPPRESSES "SEEK"

CM will now work properly for Halt/Load units which do SEEKs.

P1383 MCP - JOB SCHEDULETIME

The time at which a job enters the system (i.e., WFL has successfully compiled the job and it is ready to be inserted into a job queue) will be maintained for the job summary and the system log. The information is an addition to the fixed portion of the BOJ log record.

On a printed job summary, the following entry will be printed under the BOJ information:

"JOB ENTERED SYSTEM: <date> <time>"

P1384 MCP - MULTIPROCESSOR DUMPS

ANSWER was calling HARDWAREINTERRUPT67 to enter TAPEDUMP instead of calling the procedure at which TAPEDUMP pointed D0+3. This has been corrected.

P1385 MCP - ALLOW RECOVERY FILES AFTER "DS" OF "DCP"

An MCS could not set FILE.FILEKIND to RECOVERYFILE after the DCP was DSed. This has been corrected.

P1391 MCP - DIRECT READ FROM SCHEDULE FILE

It was possible to get a "DUMP BY FAULT IN D0 CODE" and hang a job when a direct datacom read was done that tried to read from a schedule station. This has been corrected.

P1394 MCP - "FILEKIND=FIRMWARE"

A new FILEKIND, FIRMWARE=115, has been defined. This FILEKIND will be used for diskpack controlware files in the future.

P1395 MCP - "DISCSTATUS" CHANGES

Errors in DISCSTATUS will now be reported with an appropriate error message (I/O errors can occur on testop or when trying to read buffer memeory from the control), as follows:

1. PK<nn> NO IO PATH
   No path to given disk.

2. PK<nn> I/O QUEUE BLOCKED
   I/O was cleared after 10 seconds in the queue without ever being initiated.

3. PK<nn> NO I/O FINISH
   I/O was cleared after 20 seconds; I/O was initiated but was never completed.

4. PK<nn> UNIT TYPE CODE:<xx> IS INVALID
   The 2-digit unit code <xx> read from the buffer memory is undefind.

P1396 MCP - "OL PK"

OL PK<nn> will now display the proper controller type: BX380, BX383 or BX385.

P1397 MCP - "PB" MT<NUMBER> VS. "ACMAX"

While printing a BACKUPPRINTTAPE, AUTOPUNCH would not fire up automatically if ACMAX were less than or equal to the number of printing BACKUPPRINTTAPEs. This has been corrected by charging the "PB MT#" against the APMAX and not the ACMAX.

P1401 MCP - "UA" PATH TO TAPE

"UA MT<nn> MPX<m> PATH<p>" will now test for existence of a path to MT<nn> before marking the path on-line.

P1408 MCP - "FAULTHANDLER" VS. TAGGED WORDS

It was possible for the processor to superhalt when a user program with an ON FAULT statement handled certain INVALID OP interrupts. This has been corrected.

P1415 MCP - TAPE "403" RESULT DESCRIPTOR

Under some circumstances, during tape parity retry the I/O length in a tape result descriptor could be incorrect. This has been corrected.

P1432 MCP - "CP" VS. "AIT"

Under certain conditions it was possible for RERUN to abort with an INVALID OP in blocksearch after restoring the AIT. This has been corrected.

P1433 MCP - "CP" SECURITY

If a non-privileged user job took a checkpoint with a temporary disk file open, a subsequent RERUN would be DSed for a security error. This problem has been corrected.

P1454 MCP - "READALABEL" UPDATE OF TAPE KIND

When a tape is VOLUME ADDed, the site, creationdate, tapekind, etc., fields are left 0 so that READALABEL will update them when the tape is first mounted. The TAPEKIND field is now properly updated by READALABEL.

Also, FINDINPUT will now report a backup kind of "MT" for family members (TAPES) whose kind is still 0 instead of "MT7" in the "NO FILE" message on machines with cataloging set.

P1457 MCP - FILEDATA CATALOG RESIDENT INFORMATION

A CATALOG command now reports correct resident information on cataloged files.

P1478 MCP - ENDING BANNER VS. PAGE SIZE

The "END" and "CONTINUED" print banners produced by AUTOBACKUP will no longer overlap onto a second page when a printer has small paged forms.

P1479 MCP - PROTECTED FILES VS. UPDATE FILES

Protected FILETYPE 1 disk files would give EOF NO LABEL when extended past EOF after a Halt/Load. This problem has been corrected.

P1480 MCP - REPLY LEFT ARMED "(NODISK)"

The procedure GETUSERDISK could exit leaving the reply word armed and the segments required message displayed. This has been corrected.

P1482 MCP - "SWAPPER" VS. "EI"

If SWAPPER is scheduled because of a system EI, swappable tasks may be scheduled forever. This problem has been corrected by allowing the scheduled tasks to run outside of subspace.

P1483 MCP - BLOCKSEARCH, EPILOG VS. "GOTOSOLVER"

Bad GOTOs from epilog procedures declared in blocks with no local array declarations now work properly.

P1487 MCP - "FLATREADER" BAD RECORD MESSAGE

FLATREADER now produces error messages that include the record number and type of directory record in error.

P1493 MCP - "LIBMAINT" VS. "TAPEPARITYRETRY"

A tape I/O error which occurred while writing the tape directory when copying files from the Halt/Load family would hang the system. This has been corrected.

P1512 MCP - "DMSOPEN" HEADER

DMSOPEN had a disk file header descriptor in its local stack when it got swapped out to be enlarged by PBIT on SIB. The DESCRIPTOR has been replaced by an SIRW to the header stack.

P1537 MCP - DANGLING REPLY EVENT IN "GETUSERDISK"

GETUSERDISK will no longer leave a reply event armed when it is DSed just before displaying the RSVP.

P1553 MCP - CONTROLLER MALFUNCTION

For certain I/O errors, the "I/O path" involved will be marked off-line by the MCP, after which the MCP will output the following message:

UNIT MPX<m> PATH<p> RD=<xxxxx> PATH MARKED OFFLINE

Currently, this action will be taken only for controller malfunction on diskpack controls.

Model 1 and Model 2 multiplexors paths cannot be locked out by the MCP; therefore, on these systems, the path will continue to be used.

P1554 MCP - "RCNTL"

Tape files can now have up to 2**26 = 67,108,864 records without getting record count errors.

P1555 MCP - MAKE "SWAPPER" SAVE WHEN RUNNING

SWAPPER dials the processors to handle stack stretching searches; it must, therefore, be save when it is running.

P1559 MCP - VOLUME DELETE

Deleting volumes from a volume family with more than one family member was corrupting the volume library. This problem has been corrected.

P1560 MCP - "AX"

If the operator inputs an AX command to a task in the schedule, the controller will now reply "IMPROPER STACK STATE".

P1561 MCP - "FORMATBUFFER"

The FORMATBUFFER size will now be determined by the format intrinsics rather than the MCP.

P1562 MCP - MODEL "4" MULTIPLEXOR

The B6800 Model 4 Multiplexor always "locks path" when an error occurs on pack or disk. The MCP will now unlock the path, even if the readlog command is not issued.

P1563 MCP - "IAD" VS. "HLUNIT"

IAD disk packs in the Halt/Load family may now be closed if there are no open files on them.

P1659 MCP - "TD830" VS. NON-TD830 "ODT"

The extra ETX character has been eliminated from systems which are using TD800(TD804), CONRAC as ODTs. The ETXs are required on TD830 ODTs by the terminals' internal firmware.

FILEDATA has also been corrected; it must be used in conjunction with this change.

The following hardware must be changed in some cases to cause the correct terminal type to be reported:

B6700 with SLC ODT controls:

CONRAC — The terminal ID block on the SLC needs a pin inserted in the "CON" location of each ODT.

TD800 — The terminal ID block on the SLC needs a pin inserted in the "9348" location of each ODT.

TD830 — The terminal ID block on the SLC needs a pin inserted in the "9348" and "BIDS" location of each ODT.

B6800 with SDC II ODT controls:

TD830 — These are the only ODTs connected to this control. They require no change for the use of these patches.

P1667 MCP - UPDATE "I/O" VS. "IOERROR"

The problem with update I/O to a disk file has been corrected. Formerly, if an I/O error retry occurred when writing the last record of a disk file row, and if the next row were on a different unit, the read of the next record would not be done.

P1668 MCP - MEMORY DUMP

An intermittent failure to take a memory dump to tape could occur on multiprocessor systems when a processor other than the one taking the dump gets an I/O finish interrupt before the PROC-TO-PROC interrupt issued by the dumper.

This has been corrected.

P8629 MCP - "PD" TO MISSING FAMILY

PD commands directed to missing families will now get the response "NO FAMILY".

P9024 MCP - "FIBLOCK" AND "TIMESTAMP"

1. Binary disk I/O will no longer hang.
2. READ(F,*,A) will no longer change the disk file TIMESTAMP.

P9176 MCP - "SWAPPER" QT'S "BACKUP"

BACKUP will no longer be QT'ed by SWAPPER when running in swapspace.

P9286 MCP - "DM6700" VESTIGES REMOVED

The vestigial code and external declarations for DM6700 have been removed from the MCP.

D2240 NDL - "FINISH TRANSMIT" WITH DELAY

The NDL Language Reference Manual, Form 5000953, dated 1-75, incorrectly describes the function of the FINISH TRANSMIT statement when the optional <delay time> is supplied. The complete new semantics on page 5-95 should read as follows:

The purpose of the <finish statement> is to complete the transmission of the last transmitted byte in preparation for receiving information. If the <delay time> is not specified or is NULL, the adapter cluster will wait for the completion of the transmission of the last byte and then an additional 2 milliseconds before allowing the DCP to proceed to the next statement. In addition, the line is placed in receive mode, although the adapter cluster will not sense any characters for an additional 25 milliseconds. An INITIATE RECEIVE statement should precede any subsequent <receive statement> to override the 25 millisecond delay. If the <delay time> is specified and is non-NULL, the DCP will simply perform a delay for the specified time. An INITIATE RECEIVE statement is also required in this case to place the line in receive mode.

NETWORK DEFINITION LANGUAGE

P1056 NDL - INCREMENT SEQUENCE CORRECTED

The NDL compiler was failing to emit the proper ABORT label when the INCREMENT SEQUENCE statement was used without any error action specified. The bad code then caused an INVALID INDEX fault in Pass One of DCPPROGEN when it tried to define the non-existent label. This has been corrected.

P1361 NDL - AUXILIARY LOGIC CORRECTION

In II.9 the NDL compiler was enhanced to allow selective DCP code assignment to either local or main memory; however, under certain circumstances (such as having no local memory, or using the $AUXLOGIC statement without also using the AUXILIARY statement in the DCP section), the selective code assignment was not being performed properly. The problem usually appeared as a series of "LABEL ERROR"s encountered during PASS TWO of DCPPROGEN. All known or suspected problems related to auxiliary logic usage have now been corrected.

P9121 NDL - ONE-CHARACTER REQUEST IDENTIFIERS

An NDL compiler error message of "UNDEFINED REQUEST IDENTIFIER" or "UNDEFINED CONTROL IDENTIFIER" was being generated erroneously in the TERMINAL section of a SOURCENDL program that referenced one-character length requests or control section identifiers. This problem has been corrected.

P9122 NDL - "EOF" WITH MISSING DEFINE CROSSHATCH

Under some circumstances, the NDL compiler would go into an infinite loop if the crosshatch (#) terminating a global define were missing and end-of-file on the input file were encountered. This problem has been corrected; now, the error message "INCOMPLETE DEFINITION" will be displayed when no terminating crosshatch has been found by end-of-file, and the compilation will be aborted.

PATCH

P1198 PATCH - INCORRECT SEQUENCE ERROR ISSUED

A problem with incorrect sequence errors being issued by SYSTEM/PATCH has been corrected.

P9123 PATCH - INCORRECT HANDLING OF "$.DISK" CARDS

PATCH will no longer throw away the last 80 columns of input records after processing a $.DISK command that is within a $.PATCHDECK, $.DISK$ or $.FILE command.

PLI

D2261 PLI - "LEX LEVEL" > "15"

A warning is now issued for blocks which are nested to a lex level >15.

D2422 PLI - "STATISTICS"

Times reported when the STATISTICS control card option is set are now processor times rather than elapsed times.

Because the MCP intrinsic TIME(12) is being called, more overhead is involved in reporting processor times.

PLI

P1041 PLI - FAILURE TO FLAG LARGE PRECISION

The PL/I compiler sometimes failed to flag declarations with too large precision specified. The following example was not being flagged, but was being treated as having precision (12,0):

```
DCL B BIN FIXED(780)
```

The compiler now emits a Level-3 warning for all precision values which are too large.

P1302 PLI - "RESTARTED" TASK ATTRIBUTE

The RESTARTED task attribute is now being correctly handled as a boolean task attribute. Previously, the compiler was treating the RESTARTED attribute as a file attribute.

P1467 PLI - "SET STATISTICS"

When the control card option STATISTICS was set via control cards containing quotes; i.e., "SET STATISTICS", the next card image would be lost.

Now, both "SET STATISTICS" and $SET STATISTICS work correctly.

P1468 PLI - PREPROCESSOR "INITIAL" ATTRIBUTE

The INITIAL attribute was not handled correctly under the following circumstance: The INITIAL attribute was applied to a declaration local to a preprocessor procedure.

Example:

```
   MAIN: PROC;
   %P:PROC RETURNS(CHAR);
        DCL STR CHAR INIT('I=J');
(1)     RETURN (STR);
   %END P;
   %DCL P ENTRY; /*ACTIVATE P*/
        P;
   END MAIN;
```

Statement (1) would cause an erroneous preprocessor error to be given. The INITIAL attribute is now handled correctly under all circumstances.

D2254 PLINTRN - SUMMARY OF CONDITIONS AND DS'ED DISPLAY

The information printed when a PL/I program aborts has been improved.

1. The complete STMTNO information is printed in the description of the conditions that are raised.

2. The stack history printout is expanded to a maximum of 5 lines.

3. The DS'ed display now indicates the full description of the condition.

4. Only the DS'ed display (not the complete printout) is produced for PL/I programs run from CANDE (if no dumps were taken and the options for the task do not specify a program dump).

5. File and task attribute errors (run-time) will cause an ATTRIBUTE condition (ONCODE=2000 through 2004). The system action for this is to cause the ERROR condition.

   The ATTRIBUTE condition (ONCODE=2000) has been caused in the past, but the ERROR condition was never caused. It is now possible to recover from file and task attribute errors by using an ON ERROR unit.

PLI INTRINSICS

P1099 PLINTRN - CHARACTER TO BIT CONVERSION

A character string declared CHAR(48) assigned to a bit string declared BIT(48) would result in the bit string having a value of all zeroes regardless of the value of the character string.

Example:

```
P:PROC;
  DCL B BIT(48);
  DCL C CHAR(48) INIT ('01010101');
  B=C;
END P;
```

C would be '01010101' but B would incorrectly be '0'B. This has been corrected.

P9125 PLINTRN - "ISAM" , "ISCLOSE" CHECKS VALUE OF PARAMETER

The ISAM intrinsic procedure ISCLOSE will now more carefully check the value of the second parameter "type" for being in the value 0 through 3.

If a bad value is detected, the bad parameter bit (Bit 12) will be set in the result returned from the intrinsic. The file will be closed with RELEASE (as if the type had been 0).

This change will prevent possible system failure from occurring if a bad "type" value is passed to ISCLOSE.

P9126 PLINTRN - "ISAM" , DELETE FIRST RECORD IN FILE

The ISAM intrinsic ISDELETE was not properly deleting the record with the lowest key. The key was being left in the "FINE" and "COARSE" index tables; thus, the record could still be found.

This problem has been corrected; the first record of the ISAM file will now be correctly deleted.

P9127 PLINTRN - "ISAM" , NUMERIC KEYS

The ISAM intrinsic ISDELETE will now correctly delete records when the key "mode" is one of the varieties of numeric (MODE=0, 2, 3, 4, 7 or 8).

The ISWRITE intrinsic will properly check for duplicate records (and add them when requested) when these records contain "numeric" keys.

The intrinsics were comparing record keys for exact bit pattern match in these cases. The intrinsic now uses the proper compare based upon the mode of the key; thus, numeric keyed records are correctly deleted and correctly added.

P9128 PLINTRN - "ISAM" , ADDING RECORDS TO UNBLOCKED FILES

The ISAM intrinsic ISWRITE will now properly add records to a file that was created with MAXRECSIZE the same value as BLOCKSIZE. The intrinsic was not properly reading up existing records prior to modifying their record links; this would destroy all data in the record immediately before (logically) the added record and lose the links to the record following (logically) the added record.

P9129 PLINTRN - "ISAM" , RETURN PROPER VALUE FROM "ISKEYWRITE"

The ISAM intrinsic ISKEYWRITE was erroneously returning a value of 1 when there was no error condition. The correct value of zero is now returned.

P9130 PLINTRN - "ISAM" , ERRONEOUS "ISCLOSE" RESULT OF "261"

The ISAM intrinsic ISCLOSE would sometimes erroneously return a value of 261 (rather than the value 0).

This situation would arise only if the last operation on the file (prior to calling ISCLOSE) was to add a record to the file (using the ISWRITE intrinsic, with the file opened for I/O) such that the file overflow area had to be lengthened by one block.

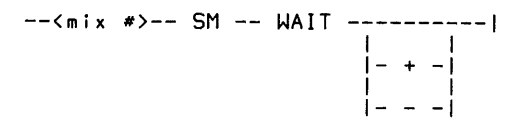
This problem has been corrected; the ISCLOSE intrinsic will now return zero in this situation.

REMOTE JOB ENTRY

D2260 RJE - "WAIT" NO LONGER A DEBUG OPTION

RJE will now allow the setting and resetting of the SM: WAIT option in versions of RJE that are not compiled with DEBUG set.

WAIT Syntax



WAIT will display current setting of option.
WAIT + will set option.
WAIT - will reset the option.

D2262 RJE - "STATIONID" CAUSED RECONFIGURATION

The RJE option STATIONID will now allow the reconfiguration of RJE stations that are not assigned to a line in the host system's NDL. The compile-time RJE defines, MAXTERMINALS and MAXTERMINALSLOG2, must be large enough to encompass as many terminals as are placed under the control of this MCS.

Note: MAXTERMINALS must be large enough to handle all RJE stations declared in the NDL, regardless of line assignments.

D2274 RJE - "SAVE/READY" STATION REMOVAL

All occurrences of saving or readying the remote site's card reader/card punch and the line printer have been removed. This will correct problems with the RJE request set while doing I/O to a unit that is being saved and or readied. This will also correct the problem of a card input file being lost when read in between to a punch output file and during the queue flush.

D2401 RJE - "REMOTEPUNCH" $ OPTION REMOVAL

There is no longer a compile-time option REMOTEPUNCH in RJE, as all versions of the symbol will be remotepunch capable.

D2406 RJE - ENHANCED BACKUP FILE HANDLING

RJE printer backup handling has been redesigned. The objectives of this redesign are to provide the following capabilities:

1. Continuation of listings after an interruption (i.e., printer jam, Halt/Load, RJE fault, etc.).

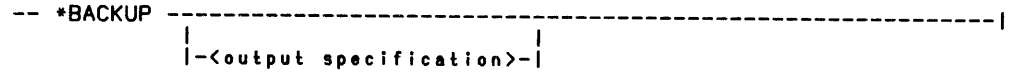
2. Ability to move printer position either ahead or back "n" pages.

3. Ability to interrogate files ready for printing and to selectively print a subset of these files.

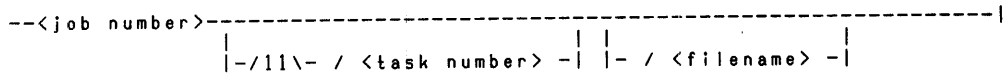
These changes only affect printer output; punch output is unaffected by the modifications to AUTOBACKUP.

A new run time option, CONTBACKUP (continue backup), has been implemented. It is set by default. Its value may be seen with the *TO command, and changed by the *RO or *SO commands. If CONTBACKUP is set and a listing is interrupted, the listing will be continued close to and before the point of interruption rather than started from the beginning. If it is reset, the listing will restart from the beginning of the current copy. A special header, with "CONT" (for continue) in block letters and the job number, will be printed before the listing is continued. Thus listings which have been continued may be easily identified. This default continuation may be overridden at print time by use of a new verb. As RJE AUTOBACKUP is printing a listing, the linkfile will be updated periodically to reflect the last page printed. When RJE recovers from an interruption, the number from the linkfile will be used to continue the listing. The RJE operator will be granted read/write access to this number. Listings may be restarted or continued on a per job basis. Punch output will not be continued regardless of the value of CONTBACKUP.

A new command, *BACKUP, has been implemented. The syntax is as follows:



<output specification>



<job number> := number between 1 and 9999 inclusive
<task number> := number between 1 and 9999 inclusive
<file name> := internal file name preceded by a 3 digit number. If the internal file name contains non-alphanumeric characters, it must be enclosed in quotes.

The BACKUP verb can be used to list all RJE printer output available, printer output associated with a particular job, or a subset of the printer output for a particular job that can be printed at the station. This verb is used to show all the files which can be output and does not show what jobs are queued for output. Printed along with the output name will be an estimate of how many pages it will take to print the specified file and, if appropriate, the form required.

For example, assume the following directory exists on disk or pack:

```
REMLP02
.       0000503
.       .       0000520
.       .       .       000CHECKS
.       .       000SUMMARY
.       0000524
.       .       0000531
.       .       .       "000REPORT FOR JOE"
.       .       000SUMMARY
.       0000545
.       .       0000563
.       .       .       000LINE
.       .       .       001LINE
.       .       0000573
.       .       .       000F
.       .       000SUMMARY
```

NOTE: "SUMMARY" files are special system files used by RJE to print headings, and will not be shown by the BACKUP verb. These are not ordinary files and will not be printed as such.

Show all printer output for this station:

```
*BACKUP
  PRINTER BACKUP
  503/520/000CHECKS,22 PAGES,FORM="CHECKS"
  524/531/"000REPORT FOR JOE",2000 PAGES
  545/563/000LINE,20 PAGES
  545/563/001LINE,25 PAGES
  545/573/000F,10 PAGES
  #
```

Show all printer backup for job 545:

```
*BACKUP 545
  PRINTER BACKUP 545
  563/000LINE,20 PAGES
  563/001LINE,25 PAGES
  563/000F,10 PAGES
  #
```

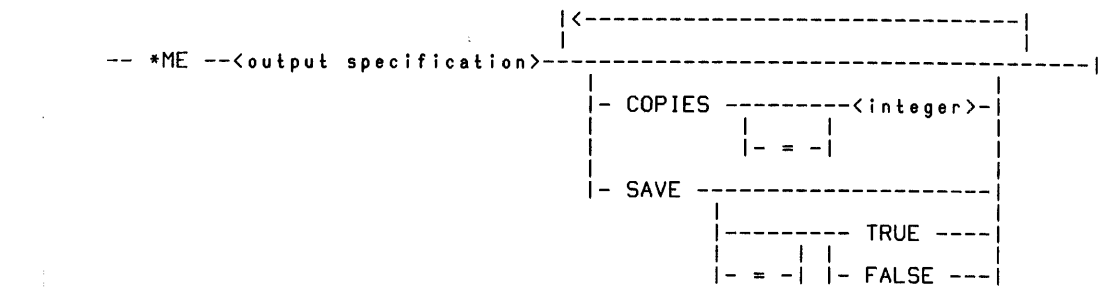
Show printer backup for subset of the entire job:

```
*BACKUP 545/563
  PRINTER BACKUP 545/563
  000LINE,20 PAGES
  001LINE,25 PAGES
  #
```

Show a specific file:

```
*BACKUP 545/563/001LINE
  PRINTER BACKUP 545/563/001LINE
  25 PAGES
  #
```

The primary use of the BACKUP command will be to display the output available so that a selective subset can be printed via the new Make Entry (*ME) command. Syntax for this command is as follows:



The *ME command is used to create a new entry in the print queue. This new entry will only cause output to be directed to the printer (i.e., will not punch any output). RJE will respond to this command with either "NO BACKUP ON DK OR PK", "JOB ACTIVE", or " YYYY.XX QUEUED". If backup is queued it will be placed at the rear of the print queue. The number (YYYY.XX) is used to distinguish between print requests on the same job. "YYYY" is the number of the job which created the printer backup and "XX" is a RJE supplied "uniqueness" number. A special header, with the word "PRINT" in block letters along with <job number>.<uniqueness number> and the *ME command that was used, will be printed before the associated files are printed. For example:

```
*ME 545 COPIES=2 SAVE
  545.1 QUEUED(ME 545 COPIES=2 SAVE)
*SP
  PRINT QUEUE
  C 503
  C 545.1(ME 545 COPIES=2 SAVE)
  #
*BACKUP 545
  PRINTER BACKUP 545
  563/000LINE,20 PAGES
  563/001LINE,25 PAGES
  573/000F,10 PAGES
  #
*ME 545/563 SAVE
  545.2 QUEUED(ME 545/563 SAVE)
*SP
  PRINT QUEUE
  C 503
  C 545.1(ME 545 COPIES=2 SAVE)
  C 545.2(ME 545/563 SAVE)
  #
```

After the backup files are printed, they will be purged unless SAVE is specified. The value of COPIES determines the number of copies of the listing that will be printed at the RJE station; the default value is 1 and the valid range is between 1 and 1023, inclusive. The use of SAVE alone will cause the files to be saved. "SAVE TRUE" and "SAVE FALSE" have the obvious meaning.

The Show Print Queue command (*SP) has been modified. Previously, this command displayed the list of jobs to be printed as they appeared in the print queue. The display has been expanded to include the following for each print queue entry:

a. If the entry will be continued after an interruption, a "C" will precede the identification number. This continuation/restart flag is set by the value of CONTBACKUP when the entry is inserted in the print queue. The value may be modified by the new "*CE" command (see below), once the entry is in the print queue.

b. If the entry is the result of a *ME command, the current specifications (i.e., the SAVE and/or COPIES values) will be used to reconstruct that command for display. This reconstructed command will be enclosed in parentheses and will follow the print queue identification number.

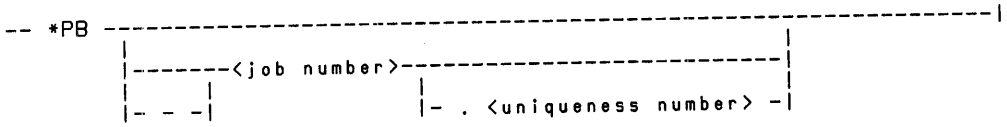
c. If the entry is not the result of a *ME command but has been modified from the default value via a *CE command, then that command will be reconstructed from the current specifications. The reconstructed command will be enclosed in parentheses and will follow the print queue identification number.

d. If the entry has ever begun to print (i.e., if it is the currently printing job or one that had been interrupted), then the display will include the file name and in parentheses the page number where continuation might occur.

For example (assuming an AUTOPRINT mix number of 200):

```
*SP
  PRINT QUEUE
  C 503
  C 545.1(ME 545 COPIES=2 SAVE)
  C 545.2(ME 545/563 SAVE)
  #
*STATUS
  #0200 AUTOPRINT/LSN038/"#0200" --FM REQD:CHECKS
  #NO TASKS ACTIVE
*200FM
*SP
  PRINT QUEUE
  C 503 FILE=503/520/000CHECKS(22)
  C 545.1(ME 545 COPIES=2 SAVE)
  C 545.2(ME 545/563 SAVE)
  C 500.1(ME 500 COPIES=5) FILE=501/000LINE(20) COPY=3
  #
```

The *PB command has been enhanced. The new syntax is as follows:



The *PB command is used to change the relative positions of entries in the print queue and possibly to create an additional entry. The semantics of each use are as follows:

*PB

When AUTOPRINT suspends itself after printing a formed backup file, "*PB" is used to resume AUTOPRINT.

*PB <job number>

This form of the *PB command retains its original meaning. It is used to cause the backup produced by <job number> to be the next backup files output to the printer.

*PB <job number>.<uniqueness number>

This form is used to move to the head of the queue entries created by a *ME command; i.e., that entry's output will be the next files printed.

*PB - <job number>
*PB - <job number>.<uniqueness number>

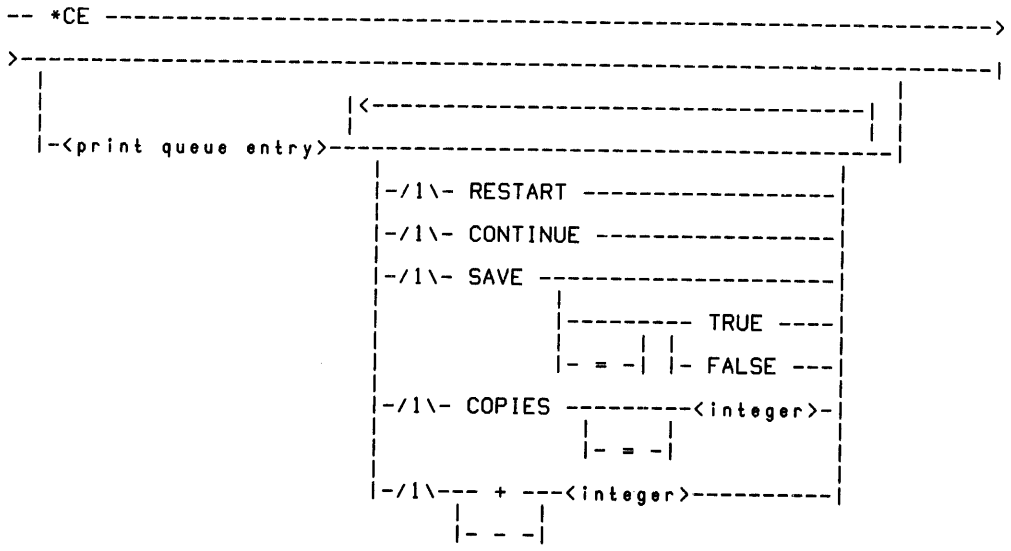
This form is used to remove a print queue entry from the print queue. The currently printing print queue entry may not be deleted.

For example:

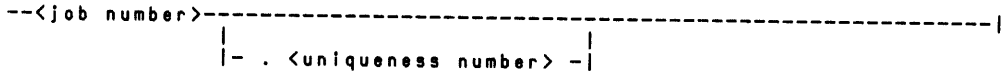
```
*SP
  PRINT QUEUE
  C 503 FILE=503/520/000CHECKS(22)
  C 545.1(ME 545 COPIES=2 SAVE)
  C 545.2(ME 545/563 SAVE)
  #
*STATUS
  #0200 AUTOPRINT/LSN038/"#0200" -- ENTER "*PB" TO CONTINUE
  #NO TASKS ACTIVE
*PB
  #

*SP
  PRINT QUEUE
  C 545.1(ME 545 COPIES=2 SAVE)
  C 545.2(ME 545/563 SAVE)
*PB 524
  #
*SP
  PRINT QUEUE
  C 524
  C 545.1(ME 545 COPIES=2 SAVE)
  C 545.2(ME 545/563 SAVE)
  #
```

```
*PB 545.2
  #
*SP
  PRINT QUEUE
  C 545.2(ME 545/563 SAVE)
  C 524
  C 545.1(ME 545 COPIES=2 SAVE)
  #

*PB -545.1
  545.1 DELETED
*SP
  PRINT QUEUE
  C 545.2(ME 545/563 SAVE)
  C 524
  #
```

With CONTBACKUP set (default value), listings will automatically be continued across an interruption. This default continuation may be changed by the new Change Entry (*CE) command. The syntax is as follows:



<print queue entry>



The *CE command is used to modify or interrogate entries in the print queue. The format of the output will be similar to that of the *SP command. The value of RESTART, CONTINUE, SAVE and COPIES has no effect on punch output. For example, if SAVE is specified for the <print queue entry>, the punch output for that entry will be output once and then the corresponding files will be removed. The printer output will be printed "COPIES" number of times and retained. RJE will verify that the *CE command will result in some physical action. The semantics associated with some of its uses are:

*CE

Returns the following information:

1. A print queue identification number. If this entry will be continued after an interruption, then a "C" will precede the number.
2. If the entry is the result of a *CE or *ME command, that command reconstructed and displayed enclosed in parentheses.
3. The file name end page number of the file currently being printed.
4. If the number of copies specified is greater than one, the number of the currently printing copy.

*CE <print queue entry>

The same information given for "*CE" will be returned for the specified job in the print queue.

*CE <print queue entry> SAVE

AUTOPRINT will not remove any files when performing the request specified by the given print queue entry.

*CE <print queue entry> COPIES=<integer>

<integer> number of copies will be printed for the specified print queue entry.

*CE <print queue entry> CONTINUE

The specified job will be continued if it is interrupted, regardless of the value of CONTBACKUP.

*CE <print queue entry> RESTART

The specified job will be marked to be restarted rather than continued, regardless of the value of CONTBACKUP.

*CE <print queue entry> + <integer>
*CE <print queue entry> - <integer>

For this form of the command to be useful, the specified job number must be marked as "continue" and have been previously interrupted. The printer position (stored by RJE) of the job will be moved ahead (+) or back (-) <integer> pages. If entry of this command causes the page number to be less than or equal to zero, RJE will continue printing at the beginning of the current backup file. If entry of the command causes page number to be greater than the number of pages in the current print file, RJE will resume printing at the next print file.

For example, assume that file 001LINE of the following backup directory is being printed:

```
545/563/000LINE
545/563/001LINE
545/573/000F
```

If an interruption occurs and the page count is decreased to less less than or equal to zero via the *CE command, printing will resume at the beginning of 001LINE. If AUTOPRINT's page count is increased past the size of 001LINE, printing will restart with 000F.

Examples of the use of *CE are:

```
*SP
  PRINT QUEUE
  C 545.2(ME 545/563 SAVE) FILE=000LINE(0)
  C 524
  #
*CE
  C 545.2(ME 545/563 SAVE) FILE=545/563/000LINE(1)
*CE 545.2 RESTART
    545.2 (ME 545/563 SAVE) FILE=545/563/000LINE(2)
*CE 545.2 CONTINUE
  C 545.2(ME 545/563 SAVE) FILE=545/563/000LINE(3)
*CE 545.2 COPIES=5 SAVE=FALSE
  C 545.2(ME 545/563 COPIES=5) FILE=545/563/000LINE(4) COPY=1
*CE 545.2 COPIES=1 SAVE
  C 545.2(ME 545/563 SAVE) FILE=545/563/000LINE(5)
*CE 524
  C 524
*CE 524 RESTART
    524
*CE 524 CONTINUE
  C 524
*SP
  PRINT QUEUE
  C 545.2(ME 545/563 SAVE) FILE=000LINE(13)
  C 524
  #
```

After a Halt/Load occurs:

```
*SP
  PRINT QUEUE
  C 545.2(ME 545/563 SAVE) FILE=000LINE(13)
  C 524
  #
*PB 524
  #
*SP
  PRINT QUEUE
  C 524
  C 545.2(ME 545/563 SAVE) FILE=000LINE(13)
  #
*CE 545.2 +3
  C 545.2(ME 545/563 SAVE) FILE=545/563/000LINE(16)
```

Care should be taken when manipulating print queues to prevent the removal of files which are required for printing at some later time. For example:

```
*ME 545
#  545.3 QUEUED(ME 545)
*ME 545/563 SAVE
#  545.4 QUEUED(ME 545/563 SAVE)
```

If no other queue manipulation is done, then when 545.4 is about to be printed, AUTOPRINT will discover that there are no output files available to be printed, because 545.3 removed all the files under its directory, thus removing the files 545.4 would have printed. To avoid this situation, either 545.4 should have been entered before 545.3 or SAVE should have been specified for 545.3. With the appropriate *ME commands and the features provided for manipulation and interrogation of the print queue, there should never be a situation where the operator cannot prevent the unwanted removal of backup files.

There is one very important point about the representation of printer position. Although communication to RJE is in terms of pages, what is stored internally is a backup record number. RJE uses approximating formulas which transform pages into backup records and vice versa. Page numbers are guides of where to start printing. RJE may not go back the exact number of pages or even start at the beginning of a page. Access could be given to the backup record number, but this is not natural and probably human transformation of pages and backup record numbers would be much less accurate than RJE's.

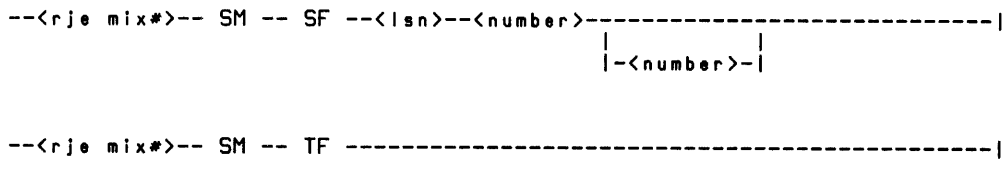
D2426 RJE - DEVICE NOT READY ABORT "QT" OF PRINTER

RJE will now QT any printer or punch file that is running to an output device that has been marked as permanently not ready (06 message from the remote site to the host MCS), instead of assuming EOT and removing the files.

D2431 RJE - IMPLEMENT "SF" AND "TF" ON A STATION BASIS

RJE will now allow the characters-per-transmission blocking factor and buffer size to be altered on a station-by-station basis. A station may set its blocking factor and buffer size by using the *SF and *TF RSC SPO command or the host site can do it with the SM commands SF and TF.

SM SF and TF Syntax:



The SF SM command allows the setting of the values for a particular station (LSN). The first number is the character-per-transmission blocking factor and the second is buffer size.

The TF SM command displays the current values of the character-per-transmission and buffer size for all active RJE stations.

There is no change to the syntax for the *SF and *TF commands at the remote site.

D2489 RJE - "LOGON" SYNTAX CHANGES AND EXTENSION

RJE will now allow LOGON in the same manner as CANDE. The USERCODE and PASSWORD may be entered on the same line with a "/" (slash) or a blank space between them. If the USERCODE and/or the PASSWORD are not supplied, RJE will prompt with either "#ENTER USERCODE PLEASE." or "#ENTER PASSWORD PLEASE."

The no password syntax is indicated by the use of the "." (period) and optionally by "*NOPASSWORD" for remote terminals that use the "." as a control character (B700 series machines).

D2490 RJE - "*PER" IMPLEMENTATION

RJE will now display the ready/not ready status of a remote station's peripheral devices. This command will only report the status of devices that the remote processor reports to the host system (e.g., not all occurrances of devices going ready/not ready are reported for most RJE systems).

```
-- *PER ------------------------------------------------------------|
```

D2491 RJE - "LOGON" WITHOUT "PASSWORD" ON "B700" SYSTEMS

B700 series systems use the "." (period) as a system control character for their RJE firmware. This make it impossible to LOGON without a password using the CANDE syntax of "." meaning no password. RJE will now allow the following substitution at LOGON time:

"*NOPASSWORD" in place of "."

D2492 RJE - BACKUP FILE ROUTING

A new method of routing BACKUP files has been implemented. Needless AUTOBACKUP tasks when CANDE output is directed at an RJE remote site have been eliminated.

D2504 RJE - "RSC" "TERM" DEFAULTS AND USE IMPLEMENTATION

The NDL values of the RJE's RSC station for LINE WIDTH and PAGE SIZE(LINES) are now used as the default value settings.

The RSC's ouput will not be affected by the setting of the TERM command's WIDTH value. This will allow the use of non-teletype type devices (72-character widths).

Note: Some RJE terminals have RSCs with a greater width per line than their internal buffersize (B700 with TC4000 as RSCS). Setting TERM WIDTH to greater than this buffersize will cause the first message greater than the buffersize to hang the remote terminal.

D2529 RJE - RJELINKED FILE TITLE CHANGE

RJE will now create its link file in the following manner:

<mcs name>/LINKFILE.

This will allow multiple copies of RJE (with different names) to run at the same time and not interfere with each other through their LINKFILEs.

Example:

SYSTEM/RJE will have the following linkfile title:
SYSTEM/RJE/LINKFILE

SYS/RJEXX will have the following linkfile title:
SYS/RJEXX/LINKFILE

D2560 RJE - "*STATUS" REPORT NO TASKS ACTIVE

RJE will no longer report "NO TASKS ACTIVE" in response to a "*STATUS" RJE request when a WFL compiler is active. The mix number of the WFL compiler will now be displayed or "NO ACTIVE WFL TASK" will be reported.

REMOTE JOB ENTRY

P1010 RJE - BLANK CARDS INSERTED INTO INPUT CARD DECKS

RJE will no longer, under certain conditions, insert blank cards into input card decks.

P1011 RJE - USERCODE ATTRIBUTES FOR "RJE" STATIONS

RJE will now set all the necessary attributes for the processed task "WFLCOMPILER". WFL no longer sets them for jobs run without a usercode and a LOGON SET TRUE RJE station.

P1012 RJE - "INVALID INDEX" FOR >15 "BD" FILES

RJE will no longer terminate with INVALID INDEX when more than 15 BD files are present for any one RJE station.

P1013 RJE - "B6700" VS. "B7700" BACKUP HANDLING

B6700 and B7700 RJE's will now agree as to the syntax of the *CE command.

P1045 RJE - FAMILY ATTRIBUTE INCORRECT SYNTAX

A usercode that does not have a FAMILY statement define in the USERDATAFILE will no longer get "FAMILY ATTRIBUTE INCORRECT SYNTAX" error messages when records are input at an RJE site with LOGON set.

P1075 RJE - EXTRA PAGE OF PAPER

RJE will no longer skip out an extra page of paper:

1. Before the EOJ banners.

2. Between the job summary listing and any following BD file.

P1121 RJE - "REBUILDPRINTQUEUE BD" FILE CORRUPTION

The RJE procedure REBUILDPRINTQUEUE will no longer corrupt BD files when a file with a LASTRECORD attribute of -1 is found.

P1122 RJE - "INVALID INDEX" WITH ONE WORD MESSAGE

RJE will no longer have its AUTOPUNCH task DS'ed with INVALID INDEX when processing a message of one word.

P1257 RJE - PRINTING JOBS WITH MORE THAN "100 BD" FILES

RJE will now handling the printing of jobs that produce more than 100 BD files at one time. It should be noted that very large directories of BD files (e.g., REMLP<xx>) will cause a deceleration of RJE during print queue rebuild and insertion.

P1303 RJE - "REMOTECONTROL" VS. "PRIMARYQUEUE"

A timing window in RJE has been corrected which allowed a REMOTECONTROL printer/punch ready/notready message to be inserted into PRIMARYQUEUE instead of BACKUPUTILQ. This would cause either an INVALID INDEX in the OUTERBLOCK or a SEG ARRAY ERROR in HANDLECARDINPUT. This has been corrected by checking for the messages as they are removed from PRIMARYQUEUE.

P1304 RJE - "USER" OPTION FOR "WFL" SECURED READER

The RJE option USER will now work correctly with II.9 WFL.

P1306 RJE - "RJELINKED" RECOVERY PROBLEM

A possible problem when the RJELINKED file is used at recovery time has been corrected. The RJELINKED file is now zeroed at creation time.

P1424 RJE - "SYSTEM/BACKUP" VS. DISK OR PACK

When SYSTEM/BACKUP is zipped (*SB, <mix#>HI, REMOTEPUNCH RESET) by RJE, the MCS will now first look for BACKUP on the family called DISK; if it does not find it on DISK or there is no family called DISK, it will then look on the family called PACK. If BACKUP is not found on DISK or PACK, an error message will be displayed to the requestor.

This will correct the problem of RJE defaulting to DISK on systems that do not have DISK.

Note: This change requires the use of II.9 or later WFL.

P1425 RJE - FOREIGN STATION TRANSFER

RJE will now update its STATIONNAME array correctly when a station not belonging to RJE is transferred to it from another MCS.

P1531 RJE - LOSING BLANK CARD AT END OF BLOCK

RJE will no longer lose a blank card of input when a single blank card (nonblank cards before and after it) falls at the end of a block during transfer to WFL.

P1558 RJE - "WFL COMPILER" WITH NO "PRIORITY"

RJE will now use the <mcs> priority *5 to fire off WFL COMPILERS when the following are true:

1. The station has LOGON set.

2. The logged-on usercode has no PRIORITY (0) declared in the USERDATAFILE.

or

1. The station has LOGON RESET.

P1566 RJE - "DESTNAME" "NEQ" "RJE" TERMINAL PROCESSOR

RJE will now handle CANDE and other tasks run outside of RJE which set DESNTAME to a member of the RJE station other than the processor.

Note: This is not a recommended way of directing output to RJE. The correct method is to specify the RJE terminal processor station name in the DESTNAME attribute.

P1624 RJE - "BREAK" ON "RJE RSC"

RJE will now save the line when a BREAK is entered at an RSC, thus correcting a problem where messages could get out of sync due to bad ARM characters.

P1625 RJE - "FORMEDLP" ARRAY VS. "REMOTEPUNCH"

The FORMEDLP array has been increased to correct an INVALID INDEX error.

P1626 RJE - MISSING "SKIP" TO CHANNEL "1"

RJE will now handle SKIPs to Channel 1 correctly, thus eliminating an overprinting problem when the SKIP is missed.

P9033 RJE - USERCODE ATTACHED TO CARD INPUT

The usercode of the processor (RJE remote site) will be attached to all card decks that do not include a user's card.

P9216 RJE - SUMMARY FILE PURGE BY *PB

RJE will no longer allow the summary file of another site to be purged by the input of a *PB <job summary>, since that summary file does not belong to the requesting site.

READER SORTER - ESPSIM

P1165 ESPSIM - HANDLE "CANT-READ" CORRECTLY

The ESP Simulator has been modified so that it will now correctly handle "UNDIGIT"; i.e., 4"A"-4"F" characters in the input string for a search function. Such undigit values would most likely be generated from CANT-READ characters in the original EBCDIC input. Previously, any attempt by the ESP Simulator to perform mask-type operations on such values would cause it to be DS'ed with an INVALID OP. It will now return a "NOT-FOUND" result instead, which is consistent with a SEARCH performed by a UCR.

READER SORTER - MIL6700

D2412 MIL6700 - IMPROVE DOLLAR OPTION HANDLING

The compiler dollar option handling has been improved to make it better able to handle unexpected and/or erroneous input.

READER SORTER - MIL6700

P1164 MIL6700 - REMOTE CYCLE AND VERSION

The compiler will now display the correct mark, level and patch number on the terminal when run from remote. The patch number printed on the top of the output listing is also now correct.

P1188 MIL6700 - "INCLUDE" DOLLAR OPTION

The INCLUDE dollar option has been changed to work the same as SCL.

P1189 MIL6700 - IMPLEMENT DOLLAR OPTION "VERSION"

The VERSION compiler option has been implemented. When compiling with NEW set and a $VERSION card appears in the symbolic, if the patch deck contains a $VERSION card the new symbolic will be updated to the version and cycle on the last $VERSION card in the patch deck.

The syntax is as follows:

```
$VERSION VV.CCC.PPP
```

where VV represents the version, CCC represents the cycle and PPP represents the patch.

P1190 MIL6700 - DELETE PROGRAMDUMP STATEMENT

The compiler no longer produces programdumps when $SET DEBUG is true.

P1272 MIL6700 - LOGICAL EXPRESSIONS

In an IF statement, when comparing the X register or Y register to a numeric, the only relations that are valid are the following:

```
X EQL 0
X NEQ 0
Y EQL 0
Y NEQ 0
```

READER SORTER - MILO

P1161 MILO - DELETE "IO" COMPLETE TIMERS

The UCR.DRIVER section of MILO will no longer produce a fatal RSP dump if a disk I/O takes longer than 1 second to complete, but it will now wait forever. The job of timing I/O's will be assumed by other code.

P1162 MILO - IMPLEMENT GENERAL "IO" TIMERS

MILO will now run timers on all RSP disk I/O operations. A fatal RSP dump will be produced if any single disk I/O takes longer than 10 seconds to complete.

P1163 MILO - REDUCE FLOW-STOPPED TIMERS

MILO contains various timers which are invoked for a UCR which is getting ahead of one or more of its files. The length of these timers has been shortened. This should result in shorter Reader/Sorter pauses; it should also increase the speed with which a restart runs.

P1166 MILO - IMPROVE DISK "R.D." LOGGING

MILO will now include the top 12 bits of the result descriptor (instead of the top 8 bits) in the disk error message passed to RSMONITOR. This will allow the TIMEOUT/EUBUSY bit to be logged as part of the disk error log entry. The RSLOG program has been changed accordingly. A problem where RSLOG would print out the wrong RSP number with the disk error printout has been corrected.

P1167 MILO - "ITEM-TYPE" IN "FLOW-STOPPED" SECTION

The special register "ITEM-TYPE" may now be interrogated from within the FLOW-STOPPED section of a UCR. The value returned will have the same meaning as in the INVALID-ITEM section. This feature will enable the UCR to discover whether flow stopped as a result of the reading of a batch ticket.

P1239 MILO - CONSOLE SWITCHES

In order to facilitate testing and debugging of the RSP software, MILO has certain run-time options built into it, any one or more of which may be invoked by raising the appropriate switches on the front panel of the RSP. To help prevent accidental use of this feature, MILO will only examine the switches if the special code of ●D● (4"D") is set up in the most significant hexade; i.e., the left-most four switches. The currently defined options are as follows:

D00001 MILO will immediately generate an error call, which will abort the operation of the RSP, and will cause a full RSP memory to be taken.

D00002 MILO will generate and log a fake disk error, and will then abort.

D00004 MILO will generate a fake queue handling error, by leaving a message in its input queue locked, and will then abort.

D00008 MILO will attempt to force a fatal memory dump on the host system anytime that it aborts.

D00010 MILO will generate an error call immediately when any running UCR aborts. This will abort the operation of the RSP, and will generate a full memory dump of the RSP.

P1249 MILO - HANDSHAKE FAILURE

The handshake procedure between B6700 and RSP has been changed. The RSP expected handshakes every 10 seconds from the B6700. If none arrived within that interval, RSP assumed B6700 had died and proceeded accordingly. There are times, however, when B6700 is busy and will not send handshakes that often; e.g., large file family rebuilding.

RSP now accepts handshakes but does not assume the B6700 is dead just because no handshakes have been received.

RSP also schedules handshakes to the B6700 every five seconds. Now RSP checks certain conditions, such as B6700 memory dump, before sending out the handshake. If these conditions are present, a handshake cannot be sent; instead, a null operation is sent through the BIC. This operation will detect a manual Halt/Load or general clear during these conditions, which will cause the RSP to breakout on its own and not wait forever to hear that the B6700 has resumed operation.

READER SORTER - READERSORTER

D2414 READERSORTER - ADD "RSC-3" TEST NAME

The name "RSC3MAINT" has been added to the list of names which may be specified as test routines files to be loaded to an RSP. This is to allow the new system file SYSTEM/RS/RSC3MAINT to be loaded. This file is the test routine which will be used in conjunction with the RSC-3 Control and B9137 Reader/Sorter.

READER SORTER - READERSORTER

P1035 READERSORTER - MODIFY "RSP DIED" RESULT DESCRIPTORS

The format of the result descriptors generated by RSMONITOR in the case of RSP failure ("RSP HARDWARE ERROR") have been changed in accordance with the general changes made for the B6800. The result descriptors will now contain a word count rather than an ending memory address.

READER SORTER - RSLOG

P1169 RSLOG - CORRECT DATE PRINTOUT

RSLOG will now print the correct time and date on the log output listing.

P1170 RSLOG - HANDLE MULTI-RECORD LOG ENTRIES

RSLOG will now correctly handle multiple segment log entries. Previously, the continuation records of these types of log entries would be treated as potential RS log entries. This would occasionally lead to program errors, specifically in the TIME&DATE routines, as the program tried to analyze unrelated data.

READER SORTER - RSNETL

P1168 RSNETL - DECLARATION OF ZERO READER/SORTER

RSNETL now generates an error message when zero Reader/Sorters are declared in Reader/Sorter Network Language. This avoids possible problems when the RSNET file is used.

READER SORTER - SORTERCONTROL

D2412 SORTERCONTRL - IMPROVE DOLLAR OPTION HANDLING

The compiler dollar option handling has been improved to make it better able to handle unexpected and/or erroneous input.

D2413 SORTERCONTRL - ALLOW "PIC X" IN READER/SORTER RECORD

The operation of the document scanner has been extended to allow for the specification of "PICTURE X" type items in Reader/Sorter record descriptions. Any elementary item described with this type of picture will, when scanned ("touched"), be allowed to contain any combination of characters. However, as the item is being scanned, a check will be made for can't-read characters, and the presence or absence of these will be used to condition the "CANT-READ" flag for the item. This feature may be used, for example, for the following purposes:

1. To check for the presence or absence of can't-read characters in a field whose actual length and/or layout is unknown.

2. To verify the validity (for length and absence of can't-read characters) of a field whose length is known but whose layout is unknown.

D2457 SORTERCONTRL - CONDITIONAL "OMIT"

The OMIT dollar option can now be conditionally set or reset based on the value of a Boolean expression, by use of the following type constructs:

```
SET OMIT = <boolean exp>
RESET OMIT = <boolean exp>
```

This functions substantially as in ALGOL, except that the only valid operators are the following: "AND", "OR", "NOT". Parentheses may be used to override the usual order of precedence.

D2473 SORTERCONTRL - IMPLEMENT "MOVE LENGTH"

The construct MOVE LENGTH is now implemented.

D2474 SORTERCONTRL - DOLLAR WARNING DEFAULT

The default of the dollar card option WARNING has been changed from RESET to SET.

D2475 SORTERCONTRL - VOLUME/SUBSET CRITERIA

When compiling an end-point set, the end-point specification must initialize volume before subset criteria can be specified.

D2481 SORTERCONTRL - BLACK BAND WARNING

Current READERSORTER controls return the "BLACK BAND" (batch ticket) indication along with the actual document data itself. The RSP operating system uses this information to invoke a special routine within the UCR to handle these types of items. Future READERSORTER controls will not return this indication until after the document has been processed, and flow is already stopped.

In order to ensure functional compatibility between the old and new controls, the following changes will be made to the software:-

(1)
The special "BLACK BAND" section currently used for handling these items will be deimplemented.

(2)
Any black-banded item will be treated as a normal valid MICR item; i.e., it will invoke the MICR-item section.

(3)
After the black-banded item has been handled, the flow-stopped section will be invoked in the normal manner.

(4)
The use of the special register "ITEM-TYPE" will be extended to allow its interrogation from the flow-stopped section, where it will report the black-band indication (in the same manner as in the "INVALID ITEM" section), if a black-banded item was one of the reasons for stopping flow.

These changes will be made with the III.1 system Release. In order to draw attention to this fact, the III.0 Release of the SORTERCONTROL compiler will produce a warning message when it compiles any "USE FOR BLACK BAND" section in the UCR. This warning message may be suppressed by resetting the "WARNING" compiler dollar option.

READER SORTER - SORTERCONTROL

P1164 SORTERCONTRL - REMOTE CYCLE AND VERSION

The compiler will now display the correct mark, level and patch number on the terminal when run from remote. The patch number printed on the top of the output listing is also now correct.

P1250 SORTERCONTRL - PICTURE FOLLOWING "SIZE"

It is now possible to declare a picture which is smaller than the preceding SIZE clause.

The compiler also detects illegal filler constructs and generates an error message.

SCR

D2246 SCR - "B6800 MCP"

New hardware interrupt literals are now recognized, and, where appropriate, logged.

The format of log records reporting B6800 processor errors or diagnostic interrupts is as follows:

Word	Contents	
0	Link	
1	Date	
2	Time	
3	Type	31:16 = 2 (LOGMAJMAINT) 15:16 = 15 (MLCONFIDENCEERROR)
4	MCPID	47:12 = System Serial # 35:12 = Mark Digit 23:12 = Mark Level 11:12 = Patch Level
5	ProcessorID, WHOI Value	
6	P1 Parameter, P1 Interrupt Parameter with Tag set to 0	
7	P3 Parameter, P3 Interrupt Parameter with Tag set to 0	
8	P2 Parameter Tag	
9	P2 Parameter, P2 Interrupt Parameter with Tag set to 0	

The interpretation of P1, P2 and P3 is based on B6800 processor specifications.

D2539 SCR - "OPTIMIZER" OPTION REMOVED

The MCP compile-time option OPTIMIZER has beem removed from the MCP and Maintenance symbolics.

SCR

P1251 SCR - "SCR MARKNO III"

The SCR MARKNO is now III in SCR printouts.

P1308 SCR - "TESTOP" RESULT FOR DISPLAY

Type bits are now decoded properly in the TESTOP result for displays.

P1309 SCR - "TESTOP" SUBTYPE

TESTOP now sets the proper unit subtype for MT and PK.

P1310 SCR - RETURN "MPXIII" IN "UNITTYPE"

The unit type is now handled correctly if the unit is a Model III Multiplexor.

P1311 SCR - ALLOW "CONTBUF" READ ON "BX385"

The CONTROLBUFFER variant is now allowed on READ on the BX385 controller.

P1426 SCR - DELETE "EXPERIMENTAL" OPTION

The EXPERIMENTAL option, which prevented SCR symbolic update if EXPERIMENTAL was not set, has been deleted.

P1495 SCR - ALLOW TRAIN TABLE LOAD

The train table now loads properly to train printers.

SCTABLEGEN

P1427 SCTABLEGEN - REDUNDANT CHARACTER CORRECTION

SCTABLEGEN will now correctly handle redundant character strings of more than 1 character.

SORT

P1398 SORT - "MERGE" VS. "UNITS=1"

The ALGOL MERGE statement will now work in the case where the input files are character-oriented (UNITS=1).

SOURCENDL

P9303 SOURCENDL - MODIFICATIONS TO "TD800" REQUEST

The SELECTTD800 request was discarding all NUL (4"00") characters in the text string and not transmitting them to the terminal. Since it is sometimes necessary to use NUL characters as padding after certain control sequences are sent to the terminal, all NUL characters will now be transmitted. the POLLTD800 request was treating the NOSPACE condition on execution of the GETSPACE statement the same as a transmission error and was reporting TERMINATE ERROR. That is different from the way other poll/select requests handle NOSPACE and is incompatible with the intended way new line CONTROL procedures operate. Accordingly, GETSPACE error handling for the TD800 was modified to make it compatible.

TABLEGEN

D2400 TABLEGEN - "TABLEGEN" NO LONGER HANDLES "ALGOL"

TABLEGEN no longer creates patches for the ALGOL compiler's internal array contents. ALGOLTABLEGEN is now used to create those patches. See ALGOLTABLEGEN note D2418 for details on how to use ALGOLTABLEGEN.

D2266 WFL - LIMIT ON NESTED SUBROUTINE DECLARATIONS

The WFL compiler will now limit the maximum number of nested subroutine declarations to ten. There may be any number of non-nested subroutines.

D2423 WFL - "AREASIZE" AND "BLOCKSIZE" ATTRIBUTES

The AREASIZE and BLOCKSIZE for files built by the DECK statement in WFL have been changed:

	AREASIZE	BLOCKSIZE
BCL	1512	420
EBCDIC	1008	420
BINARY	756	480

D2429 WFL - "NEW" WFL SYNTAX CLARIFICATION

WFL system notes that discuss syntax refer to "new" II.9.0 syntax as described in II.9.0 WFL note D2077. Any other level of syntax will be explicitly described.

D2433 WFL - "LINEINFO" FOR JOB CODE FILES

Job code files now contain LINEINFO information relating code addresses back to the line number shown in the WFL statement listing. The line number will be shown in the job summary for any WFL statement that causes the job to be abnormally terminated.

D2434 WFL - CLEARING CARD READER IN USE BY "WFL"

The WFL compiler will now terminate more cleanly when the card reader it is using is CLEAR'd by the operator. An error message will be generated and the WFL compilation aborted. The job summary for the job being read will be printed.

D2449 WFL - EXPRESSIONS IN "COMPILE" STATEMENTS

In all previous versions of the WFL language, expressions have been disallowed when specifying either file or task attributes to the "compilee" in a COMPILE statement.

It will now be possible to use expressions in these contexts (but not in pre-II.9 WFL syntax). These expressions will be evaluated prior to initiating the COMPILE and the obtained values will be stored in the code file (as if they had been specified as constants).

Assigning a WFL global file to the "compilee" is still disallowed.

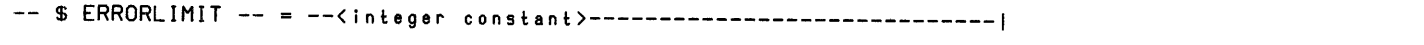
Example:

```
SUBROUTINE SUB(STRING S, REAL R);
BEGIN
   COMPILE #S FORTRAN LIBRARY;
   FORTRAN FILE TAPE=SOURCE/#S;
   FORTRAN FILE CARD=PATCH/#S ON DISK;
   PRIORITY=R;
END;

SUB ("PROD1", 55);
SUB ("PROD2", 45);
```

D2495 WFL - "$ERRORLIMIT" OPTION

The ERRORLIMIT dollar option allows control of the termination of a job's compilation due to syntax errors.

Syntax:



Compilation of the job will be terminated if the number of errors detected becomes greater than or equal to the <integer constant>. If no $ERRORLIMIT card appears, the default error limit is 99999 unless the job was STARTed through CANDE, in which case the default error limit is 10.

As with other dollar option cards, the dollar sign may appear in column 1 or column 2 of the card image. If the dollar sign is in column 2 and NEWSOURCE has been specified, that card image will be written out to the new source file; otherwise, it will not be written out.

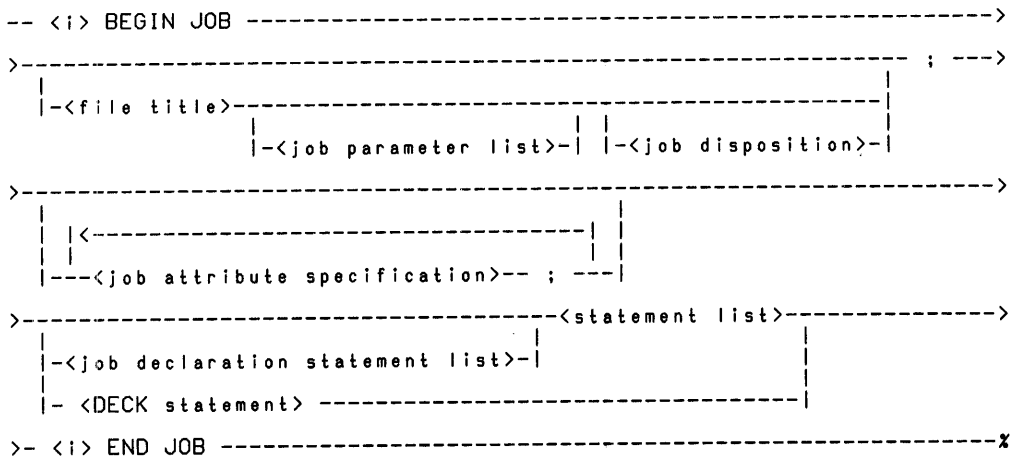
The dollar option card may not appear within internal data decks, as it will be treated as an input data record.

D2514 WFL - PARAMETERS TO JOBS ALLOWED

A WFL job may now have boolean, integer, real and string parameters. These parameters are declared in the BEGIN JOB statement. Page 8-42 of the Work Flow Language Reference Manual (Form No. 5001555) describes the passing of actual parameters in START statements. The following describes their use in BEGIN JOB statements.

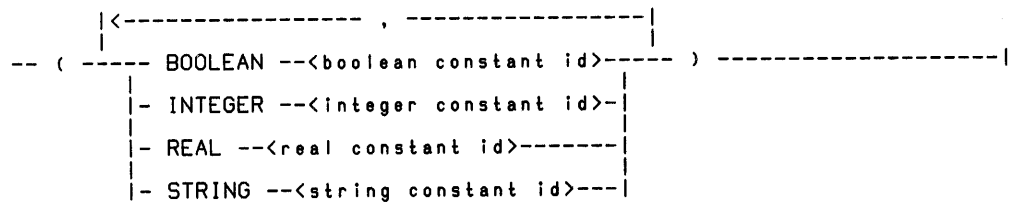
The revised syntax for <job> on page 5-1 of the manual is as follows:

<job>



The following syntax and semantics should be added to the description of JOB STRUCTURE on pages 5-1 to 5-2 of the manual.

<job parameter list>



A <job parameter list> declares named constants (not variables) of the type specified in the parameter list. A named constant of the appropriate type may be used in any context where <boolean constant>, <integer constant>, <real constant> or <string constant> is allowed, including within the <job attribute specification>.

A WFL job with parameters that is presented to the system from any source that does not allow the actual parameters to be given (such as SITE or RJE readers, system SPOs or ZIP statements of the other programming languages) must be compiled for SYNTAX. If it is also compiled with the NEWSOURCE option, a START statement in a subsequent job from any source (including the START command in CANDE) can pass the proper parameters and start the job.

The following replaces the example on page 5-3 of the manual.

Example of a Full Job

```
<i> BEGIN JOB EXAMPLE (STRING TESTNAME, INTEGER TESTNUMBER);
    NAME=EXAMPLE/#STRING(TESTNUMBER,*);
    USERCODE = WFL/MANUAL;
    CLASS = 2;

    TASK TCOMP, TRUN;
    COMPILE #(TESTNAME & STRING(TESTNUMBER,*))[TRUN] PL/I[TCOMP];
    PL/I DATA
         P: PROC;
              DCL I INIT(15);
              DISPLAY ('P IS RUNNING');
              DISPLAY ('NOW ABORT');
              I=I/0;
         END P;
<i> IF TCOMP IS COMPILEDOK THEN
         DISPLAY "COMPILED OK"
    ELSE ABORT "*DID NOT COMPILE";
    IF TRUN IS COMPLETEDOK THEN
         DISPLAY "RAN OK"
    ELSE ABORT "*RUN ABORTED";
<i> END JOB
```

D2528 WFL - TASK ATTRIBUTE "LOCKED"

LOCKED is a Boolean task attribute. Its value is not used by the MCP. The user may set it or read it at any time for his own use.

D2540 WFL - MULTIPLE FAMILY SPECIFICATIONS

A job (or task) has only one family specification associated with it. As with other task attribute specifications, the occurrence of a second family specification is syntactically correct; however, it overrides the first occurrence.

D2541 WFL - STRING PRIMARIES IN FILE TITLES

<string primary>s in <file title>s have not yet been implemented for <libmaint file list>s.

D2542 WFL - MAXIMUM STRING CONSTANT LENGTH

The Work Flow Language Reference Manual (Form 5001555), page 2-8, states that 80 characters is the maximum length of a string constant. It should be corrected to state that 256 characters is the maximum length.

The string constant may not be broken across a card boundary; therefore, for most cases, the actual maximum length is smaller than 256.

D2543 WFL - USE OF THE "STRING" FUNCTION

The Work Flow Language Reference Manual (Form 5001555) should state that the STRING function returns the rightmost characters if the integer being converted to a string has more characters than the length specified.

The following sentence should be added to the description of the STRING function on page 4-10 of the manual:

"If the value of the second <integer expression> is less than the number of characters needed to represent the first <integer expression>, the rightmost characters are returned."

The following example should be added:

```
    STR1:=STRING(1234,3);          Result = "234"
```

D2586 WFL - TEST FOR "WFL" DATA DECKS BEING RESIDENT

Testing for a data deck being RESIDENT (or PRESENT in pre-II.9 WFL) will now give correct results.

The following example indicates how to test for a data deck being RESIDENT:

```
   <I> BEGIN JOB RESIDENT/EXAMPLE;
       DATA X
         .
         .
         .
   <I>
       FILE XF(KIND=READER,TITLE=X);
       IF XF IS RESIDENT THEN;          % "XF IS RESIDENT" WILL BE TRUE
       IF FILE X IS RESIDENT THEN;      % "FILE X IS RESIDENT" REALLY
                                        % MEANS "FILE X ON DISK IS
                                        % RESIDENT", WHICH WILL
                                        % PROBABLY BE FALSE
         .
         .
```

```
<I> END JOB
```

WORK FLOW LANGUAGE

P1046 WFL - EXPLICIT LOOPS AROUND "COMPILE" STATEMENTS

The WFL compiler will now properly pass the SEGZERO parameter to compilers invoked via the COMPILE or BIND statement. The effect of this is that, if the COMPILE statement is coded within a loop, the attribute equation information will properly be passed on to all programs compiled, rather than only the first.

Partial Job Example:

```
    REAL I;
    LOOP:    COMPILE X FORTRAN;
             FILE FILE5=P ON DISK;
             FORTRAN DATA
                  .
                  .
                  .
?            I:=I+1;
             IF I LSS 5 THEN GO LOOP;
```

Previously, the equation for FILE5 was only passed to the first compilation of program X. Now it will be passed to all 5 compilations of X.

P1047 WFL - GENERATE ERROR FOR BAD "KIND" SPECIFICATION

The WFL compiler was not properly generating syntax errors for erroneous KIND values when these KIND values were not preceded by an explicit KIND=. File equations such as "FILE X (PRINTER BACKUP TAPE)" were being allowed (although the resulting KIND was TAPE).

The WFL compiler will now properly note this as an error.

P1048 WFL - "WFL" ABORTS FOR EXTREMELY LARGE JOBS

The WFL compiler will no longer terminate with a SEG ARRAY error when presented with an extremely large WFL job. The WFL compiler will now detect this situation and issue the message "JOB CODE FILE CAPACITY EXCEEDED".

P1049 WFL - STRING EXPRESSIONS IN "ON" STATEMENTS

The WFL compiler will now properly generate expression temporaries for string expressions that occur within the body of an ON statement.

P1050 WFL - HANDLING OF "KIND" LISTS IN "OLD" "WFL"

KIND lists containing BACKUP following an OR will now generate the correct KIND speficiation.

Example:

```
    KIND=PRINTER OR BACKUP PETAPE
```

Previously, this would result in a KIND of PETAPE. The proper kind is now generated (PRINTER). Note that this syntax is valid only for "old WFL" syntax.

Also, KIND lists containing either of the mnemonics PRINTER or PUNCH immediately following the token OR were being issued erroneous syntax errors by the II.9.0 system release WFL. This has been corrected.

P1051 WFL - "DATA" OR "END" WITHIN FILE NAMES

The WFL compiler will no longer be sensitive to the tokens DATA or END when they occur within file names. The WFL compiler would previously become confused if one of these tokens immediately followed the usercode part in an otherwise valid file name.

The following example will now compile correctly:

```
    .
    COPY A,B,(C)DATA/E TO T;
    . . .
```

P1052 WFL - HANDLING OF MISSING "?END JOB" CARD

The WFL compiler will again properly abort the compilation of a job when it encounters the beginning of the next job (i.e., the ?END JOB card is missing). This feature existed prior to the II.9.0 system release.

P1064 WFL - IMPROVE HANDLING END OF FILE

The WFL compiler now behaves more sensibly upon reaching End Of File on a disk file (from ZIP with file or CANDE start).

An UNEXPECTED END OF FILE message will not be produced if either:

a. The job was being "flushed" because of a severe error message, or

b. The compiler was skipping blank cards following the last job in the file.

P1065 WFL - SINGLE "START" STATEMENT

The WFL compiler will now properly handle a START statement when that statement was the only statement in the job. Previously, the WFL compiler would fault terminate for INVALID OP.

P1066 WFL - LONG "COPY" STATEMENTS

The WFL compiler would terminate for INVALID INDEX or STRING PROTECT when presented with a COPY statement containing more than (approximately) 150 file names.

This problem has been corrected; very long COPY statements are now allowed, just as in pre-II.9 WFL.

P1067 WFL - ELIMINATE "WFL" ABORT ON BAD FILE SYNTAX

The WFL compiler was aborting for INVALID INDEX for the following:

```
COMPILE X FORTRAN;
FORTRAN FILE CODE (* SECURITYTYPE=PUBLIC);  %There should be a comma
                                            %following the asterisk
.
.
.
COPY Y TO TAPEY;                            %WFL aborted here
```

The WFL compiler will now give the correct error message for the bad file specification and will no longer abort on the COPY statement.

P1068 WFL - INFINITE LOOP FOLLOWING SECURITY ERRORS

The WFL compiler will now issue an "UNEXPECTED END OF FILE" message if it detects a security error while reading a disk file. Such a situation occurs by using certain combinations of changing a tasks's usercode and performing ZIP WITH FILE statements.

Previously, the WFL compiler would loop issuing a "COMPILATION ABORTED" message.

P1376 WFL - ERRORS IN DECLARATIONS

WFL is now able to properly handle errors in declarations.

P1485 WFL - BOOLEAN CONSTANT EVALUATION

WFL was not evaluating the following expressions correctly:

```
TRUE OR TRUE
TRUE AND TRUE
TRUE EVQ TRU
```

They now give the expected results.

P1488 WFL - SINGLE BLANK CHARACTER STRING CONSTANTS

WFL now handles a string constant that is a single blank character.

P1489 WFL - CORRECT "HEX" FUNCTION

The HEX string conversion function incorrectly converted hexidecimal strings containing "A" through "F". This has been corrected.

P1658 WFL - BACKWARD BRANCHES IN "ON" STATEMENTS

Incorrect code is not longer generated when an ON statement does a GO TO to a label that occurred before the ON statement.

P1660 WFL - "$INCLUDE" WITH "NEWSOURCE" OR "LIBRARY"

When doing a WFL compile with NEWSOURCE or to LIBRARY, $INCLUDE now works as documented. If the dollar sign is in column 1, the included text, but not the dollar card, is written to disk. If the dollar sign is in column 2, the dollar card, but not the included text, is written to disk.

Previously, neither the included text nor the dollar card was written to disk if the dollar sign was in column 1. Both the included text and the dollar card were written to disk if the dollar sign was in column 2.

P9131 WFL - "WFL" TERMINATION IF "DS" WHILE SCHEDULED

The WFL compiler will now terminate cleanly if a ZIP statement (executed by some user program) is DS'ed while still scheduled.

Previously, the WFL compiler would issue two messages "WFLCODE REQUIRES MT7 #1"; this no longer occurs.

P9132 WFL - ERROR FOR MISSING "UNTIL"

The generation of an error message for a missing UNTIL in a DO statement was lost for the II.9 WFL syntax. That error message has been restored.

P9133 WFL - ELIMINATE "INVALID INDEX" ON "WFL" ABORTS

The WFL compiler would terminate for INVALID INDEX if there were a fatal syntax error in the job heading and the DISKLIMIT or TASKLIMIT job attributes had been specified. This termination occurred in such a fashion that a job printout would never be routed to an RJE station as it should have been.

This problem has been corrected; now, the WFL compiler will not fault in this situation and the syntax error listing will be routed to the appropriate destination.

P9134 WFL - PROPER INFORMATION FOR LONG "COPY" STATEMENTS

The WFL compiler will now properly use a 16-bit field (rather than only the low 13 bits) in the parameter array to LIBRARY/MAINTENANCE.

This will allow Library Maintenance to properly perform the very long COPY statements that WFL allows without terminating for INVALID INDEX.

P9135 WFL - PERFORM JOB ROLLOUT IN MORE INSTANCES

The WFL compiler will now mark in a few more instances jobs that have no active tasks.

The following jobs will be marked as having no active tasks, if the job has no active tasks and either:

1. Attempts to open a file, or

2. Waits for an operator OK, or

3. Executes the ACCEPT function, or

4. Finishes execution of any WAIT statement, or

5. Exits from a subroutine, or

6. Finishes all tasks and attempts to go to EOJ.

This means that should a system Halt/Load occur during any of the above situations, the preceding task will not be restarted (the job will restart at one of the above places).

P9159 WFL - INFORMATION CARRY OVER

If two COPY statements used the same task identifiers, various information about the sources and destinations could (erroneously) be carried over from the first COPY statement to the second.

The following example would copy Y from the pack named P:

```
COPY X FROM P(PACK) TO DISK[T];
COPY Y FROM PACK TO DISK[T];
```

This problem has been corrected; the task identifier will no longer carry source or destination information between COPY statements.

P9247 WFL - "DECK STATEMENT"

The WFL compiler will now properly handle the new DECK statement syntax.

```
? DECK  <file title>
? DATA
   ...
```

The WFL compiler will also correctly handle DECK statements in WFL jobs that do not have an explicit NAME.

```
? BEGIN JOB
? USER = X/Y
? DECK   ...
```

On the II.9 Release, both of these situations failed to lock a created disk file.

XREF ANALYZER

D2526 XREFANALY - CHANGE "INTERACTIVEXREF" FILE TITLES

The format of the titles of the files generated for INTERACTIVEXREF has been changed to the following:

<usercode>XREFFILES/<codefile title>/DECS
<usercode>XREFFILES/<codefile title>/REFS

D2527 XREFANALY - VERSIONS FOR "INTERACTIVEXREF" FILES

Version information is now included in the XREFFILES. INTERACTIVEXREF checks the compatibility of the XREFFILES and displays an appropriate error message if the XREFFILES were created with an incompatible XREFANALYZER.

XREF ANALYZER

P1429 XREFANALY - CORRECTION DECLARATION COUNT

An XREF of a symbolic which does a LOADINFO will now produce a correct count of the number of declarations. The incorrect count previously generated would result in an "EOF NO LABEL" error when attempting to generate INTERACTIVEXREF files.

P1647 XREFANALY - CORRECT "XREF" OF LARGE PROGRAMS

The XREFANALYZER will now handle very large programs without errors. Several fields have been increased to allow the storage of more information.

SYSTEST - B6800MDL/TAPEHANDLER

P1270 TAPEHANDLER - LIST CARD ASSEMBLY PART NUMBER

The system now lists card assembly part numbers correctly.

P1440 TAPEHANDLER - DELETED "FAMILY" STATEMENTS

The family statement "FAMILY DISK=<storage device family name>" has been deleted from TAPEHANDLER; consequently, job decks must be created with proper and correct family statements in order to run TAPEHANDLER.

P1448 TAPEHANDLER - NEW DATA FILE TITLE = OLD FILE TITLE

If TAPEHANDLER is asked to update a promwriter tape and the system crashes before the task is complete, the restarted task gets "EOF NO LABEL" on the file with the internal name "NEW" and title "PROMWRITER/MERGED/BOSROMFILE".

Specifications for areasize and areas file attributes have been added to the declaration for file "NEW" so that the old copy of the file will not be used on restart; this old file will be removed when the final execution of the program locks the "new" file.

The areas (=5) and areasize (=4470 records/area) are such that (1) a record of maxrecsize 149 words is never split across an area boundary, and (2) the file is large enough to hold the equivalent of a full 2400-foot reel of unblocked records when written at 2600 bits/inch with 3/4-inch IRG.

Blocksize of 149 X 30 = 4470 words is also specified. Areasize is also an integral multiple of the areasize for ALGOL symbolic files.

P1677 TAPEHANDLER - MAINTENANCE KIT RELEASE VERSION

Several new features have been added for the latest B6800 Maintenance Kit.

P1704 TAPEHANDLER - UPDATE COMMENTS IN SYMBOLIC

All the job deck examples in the comments section of the TAPEHANDLER symbolic have been updated to reflect the changes made to TAPEHANDLER.

SYSTEST - DCP/MAINTMCS

D2479 MAINTMCS - DATACOM-TO-DISK SYNTAX

The DISKTEST EU command is used to designate the diskfile electronics unit (1C3 disk) where a file will be created for use in testing datacom-to-disk controls (DCDC). This command must have been completed prior to running DCDC test 10 (i.e., TESTFILE=SYSTEST/DCP/DCDC). The "DISK" specification with a "DCPTEST" command will cause the file address, including electronics unit (EU), of the current disk file to be passed to the DCP when initiating a given test. The "DISK" specification is required for any selection which includes DCDC test 10 (and ignored by other DCP tests).

The following example will cause MAINTMCS to create a special test file on EU32 (if EU32 is 1C3 type disk). Tests 0-10 of SYSTEST/DCP/DCDC will each be passed the address of the file.

```
<mixno> SM DISKTEST EU=32
<mixno> SM TESTFILE=SYSTEST/DCP/DCDC
<mixno> SM DCPTEST DCP=0, TEST=0-10, TIMEOUT=3600, DISK; STATUS
```

D2480 MAINTMCS - "CANCEL" COMMAND FOR "DCPTEST"

The CANCEL command may be used to discontinue a DCP test if one is in progress, and to eliminate unwanted tests which have been queued. If a test has timed out and is awaiting "AX OK", "AC CANCEL" may be used instead.

The CANCEL command may be entered via card when initiating MAINTMCS or at any time when MAINTMCS is running via an "SM" from the operator display terminal. The following example shows a CANCEL followed by another command:

```
<mix no> SM CANCEL; DCPTEST DCP=0, TEST=3-5, REPEAT=100
```

D2582 MAINTMCS - FRONT-END-CONTROL SELECTION

The "DCPTEST" syntax test specification list will now allow a selection of front-end-control adapter address (i.e., basic control number and position within the basic control), AA, to be passed to the DCP as a parameter with the code file of the next test specified. (AA will be ignored by the next test specified if AA is not needed.)

Example:

```
<mixno>SM DCPTEST DCP=2,TEST=(AA3:1)4-6,(AA0:2)0-8,10
```

This example will cause DCP 2 to be selected, and would run tests 4 through 6 using the front-end-control in position 1 of basic control 3. It would then run tests 0 through 8 and 10 using the front-end-control in position 2 of basic control 0.

Additional parameters allow specification of disk electronics units (EU) or storage units (SU) for use with SYSTEST/DCP/DCDC.

Example:

```
<mixno> DCPTEST DCP=0,TEST=(AA0:0,EU31,SU0)0-9,(DISK)10-11
```

SYSTEST - DCP/MAINTMCS

P1271 MAINTMCS - IMPROPER TEST NUMBER SELECTION

Using DCPTEST syntax, the first test number of a range of tests to be executed was being ignored under certain conditions. This would typically cause "TEST 0" to be executed in place of the first test of the range. This has been corrected.

SYSTEST - IO/IOP

P1118 IOIOP - ACCELERATE DATA COMPARE

Data compare has been accelerated.

SYSTEST - OFF/CPU

P1377 CPU - REMOVE "BCL" CHARACTERS

BCL characters are no longer used for the following:

GEQ
LEQ
NEQ
:=
*

P1703 CPU - "OFFLINE" DOLLAR OPTION

The OFFLINE dollar option has been added, which forces the value array "DATAARRAY" to be a "SAVE" array.

SYSTEST - OFF/DFO

P1377 DFO - REMOVE "BCL" CHARACTERS

BCL characters are no longer used for the following:

GEQ
LEQ
NEQ
:=
*

SYSTEST - OFF/MPX

P1377 MPX - REMOVE "BCL" CHARACTERS

BCL characters are no longer used for the following:

GEQ
LEQ
NEQ
:=
*

SYSTEST - OFF/PEP

P1138 PEP - LOAD TRAIN TABLES

All the train tables contained in LTTABLEGEN can now be loaded.

P1260 PEP - CODE SPACE IN "SEG1"

More code space has been made in SEG1 for program code.

P1261 PEP - "READ EXTENDED RD OPTION"

READ EXTENDED RD OPTION has been added to the error options on flexible disk.

P1262 PEP - RUN WITH "PASS" SET

Run of test cases with PASS set has been corrected.

P1263 PEP - PERIPHERAL UNIT SEARCH

Disk pack type 235 is now found correctly in a peripheral unit search.

P1377 PEP - REMOVE "BCL" CHARACTERS

BCL characters are no longer used for the following:

GEQ
LEQ
NEQ
:=
*

P1670 PEP - "64" MEMORY MODULES

The system no longer hangs in a memory search loop if 64 memory modules are present.

P1671 PEP - CHANGE PRINTER AND DISPLAY MESSAGES

The printer and display messages have been changed to print PK3 instead of PK2 when disk pack 235 test cases are run.

P1672 PEP - SPECIFIED AND NONSPECIFIED TRAIN TABLES LOADED

Train tables are now loaded correctly when specified (by user display message) and when not specified (use testop train id).

P1673 PEP - MULTIPLE PROCESSORS INITIALIZE

Non-halt-load processors now initialize correctly on a multiple processor system.

P1674 PEP - INDICATE "PK2" TEST CASE MODIFIED

When a disk pack type 235 (PK3) test case error occurs, the system now indicates that the PK2 test cases are modified for PK3 at execution time.

P1675 PEP - NO "PB MAE" TEST IF "MM63" IS PRESENT

The pseudo-busy (PB) memory address error (MAE) test in SYSCON is now skipped if memory module 63 is present.

P1676 PEP - MAKE "RUNHCT" RUN IN CONTROL STATE

RUNHCT now runs in control state so that predicted result descriptors are correct on error retry.

SYSTEST - OFF/DISK

P1117 DSK - TITLE CHANGE

The title SYMTEST/OFF/DISK and SYSTEST/OFF/DISK has been changed in the symbolic file and the SYSTESTS tape to SYMTEST/OFF/DSK and SYSTEST/OFF/DSK, respectively.

P1377 DSK - REMOVE "BCL" CHARACTERS

BCL characters are no longer used for the following:

GEQ
LEQ
NEQ
:=
*

SYSTEST - OFF/TAPE

P1116 TPE - TITLE CHANGE

The title SYMTEST/OFF/TAPE and SYSTEST/OFF/TAPE has been changed in the symbolic file and the SYSTESTS tape to SYMTEST/OFF/TPE and SYSTEST/OFF/TPE, respectively.

P1377 TPE - REMOVE "BCL" CHARACTERS

BCL characters are no longer used for the following:

GEQ
LEQ
NEQ
:=
*

SYSTEST - OFF/DISKEX

P1377 DISKEX - REMOVE "BCL" CHARACTERS

BCL characters are no longer used for the following:

GEQ
LEQ
NEQ
:=
*

SYSTEST - OFF/MEMORY

P1377 MEMORY - REMOVE "BCL" CHARACTERS

BCL characters are no longer used for the following:

GEQ
LEQ
NEQ
:=
*

SYSTEST - OFF/MEMREQ

P1377 MEMREQ - REMOVE "BCL" CHARACTERS

BCL characters are no longer used for the following:

GEQ
LEQ
NEQ
:=
*

SYSTEST - OFF/DFOSUTEST

P1377 DFOSUTEST - REMOVE "BCL" CHARACTERS

BCL characters are no longer used for the following:

GEQ
LEQ
NEQ
:=
*

SYSTEST - OFF/TIMERTEST

P1377 TIMERTEST - REMOVE "BCL" CHARACTERS

BCL characters are no longer used for the following:

GEQ
LEQ
NEQ
:=
*

SYSTEST - SCR/PKXD

P1442 PKXD - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PK04

P1442 PK04 - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PK05

P1442 PK05 - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PK06

P1442 PK06 - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PK07

P1442 PK07 - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PK08

P1442 PK08 - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PK09

P1442 PK09 - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PK10

P1442 PK10 - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

MARK 3.0

SOFTWARE IMPROVEMENTS NOTES (P NOTES)

SYSTEST - SCR/PK11

P1442 PK11 - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PK15

P1442 PK15 - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PK16

P1442 PK16 - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/SC01

P1227 SC01 - "TD830" AND "B7700" UPDATE

MATII has been updated to run on TD830 display as well as TD800 and BIDS. The title has been changed to B6700/B7700.

SYSTEST - SCR/SC02

P1227 SC02 - "TD830" AND "B7700" UPDATE

MATII has been updated to run on TD830 display as well as TD800 and BIDS. The title has been changed to B6700/B7700.

SYSTEST - SCR/SC03

P1227 SC03 - "TD830" AND "B7700" UPDATE

MATII has been updated to run on TD830 display as well as TD800 and BIDS. The title has been changed to B6700/B7700.

SYSTEST - SCR/SC04

P1227 SC04 - "TD830" AND "B7700" UPDATE

MATII has been updated to run on TD830 display as well as TD800 and BIDS. The title has been changed to B6700/B7700.

SYSTEST - SCR/SC05

P1227 SC05 - "TD830" AND "B7700" UPDATE

MATII has been updated to run on TD830 display as well as TD800 and BIDS. The title has been changed to B6700/B7700.

MARK 3.0

SOFTWARE IMPROVEMENTS NOTES (P NOTES)

SYSTEST - SCR/SC06

P1227 SC06 - "TD830" AND "B7700" UPDATE

MATII has been updated to run on TD830 display as well as TD800 and BIDS. The title has been changed to B6700/B7700.

SOFTWARE IMPROVEMENTS NOTES (P NOTES)

SYSTEST - SCR/SC07

P1227 SC07 - "TD830" AND "B7700" UPDATE

MATII has been updated to run on TD830 display as well as TD800 and BIDS. The title has been changed to B6700/B7700.

SYSTEST - SCR/SC08

P1227 SC08 - "TD830" AND "B7700" UPDATE

MATII has been updated to run on TD830 display as well as TD800 and BIDS. The title has been changed to B6700/B7700.

SYSTEST - SCR/SC09

P1227 SC09 - "TD830" AND "B7700" UPDATE

MATII has been updated to run on TD830 display as well as TD800 and BIDS. The title has been changed to B6700/B7700.

SYSTEST - SCR/SC10

P1227 SC10 - "TD830" AND "B7700" UPDATE

MATII has been updated to run on TD830 display as well as TD800 and BIDS. The title has been changed to B6700/B7700.

SYSTEST - SCR/SC11

P1227 SC11 - "TD830" AND "B7700" UPDATE

MATII has been updated to run on TD830 display as well as TD800 and BIDS. The title has been changed to B6700/B7700.

SYSTEST - SCR/SC12

P1227 SC12 - "TD830" AND "B7700" UPDATE

MATII has been updated to run on TD830 display as well as TD800 and BIDS. The title has been changed to B6700/B7700.

SYSTEST - SCR/CONFIG

P1265 SCRCONFIG - ADD CONTROLWARE LEVEL

The controlware level has been added to the printout and MPXIII handling.

P1267 SCRCONFIG - ADD "DPK235"

The system can now run with 235 and/or 225 disk packs.

SYSTEST - SCR/PKDUMP

P1442 PKDUMP - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PKSCAN

P1264 PKSCAN - FAULT PRINTOUT

The fault printout has been expanded to 6 hex digits.

P1265 PKSCAN - ADD CONTROLWARE LEVEL

The controlware level has been added to the printout and MPXIII handling.

SYSTEST - SCR/PKSEEK

P1442 PKSEEK - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PKBASIC

P1442 PKBASIC - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/SUPERIV

P1265 SUPERIV - ADD CONTROLWARE LEVEL

The controlware level has been added to the printout and MPXIII handling.

P1436 SUPERIV - ERROR RETRY FOR NEW CONTROLWARE

Error retry for the new controlware now checks the balance of the cylinder with reads instead of verifying by track.

P1437 SUPERIV - BALANCE OF TRACK READ LENGTH

The balance of track read length is now less than or equal to track length.

SYSTEST - SCR/PKHDOVER

P1442 PKHDOVER - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PKINTERCH

P1442 PKINTERCH - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PKACTUATOR

P1442 PKACTUATOR - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PKHEADISOL

P1442 PKHEADISOL - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/PKWRITEREAD

P1442 PKWRITEREAD - NEW CONTROLWARE, DISK PACK TYPE "235"

The system now properly handles new controlware (2.0 or later) and disk pack type 235.

SYSTEST - SCR/TP72

P1496 TP72 - "SCR" TESTS FOR TRAIN PRINTERS

SCR tests have been implemented for the following train printers:

TP72	EBCDIC72 (1100 LPM)
TP96F	EBCDIC96 (1100 LPM)
TP64	EBCDIC64 (450/750 LPM)
TP64A	EBCDIC64A (450/750 LPM)
TP96S	EBCDIC96 (450/750 LPM)

SYSTEST - SCR/TP96F

P1496 TP96F - "SCR" TESTS FOR TRAIN PRINTERS

SCR tests have been implemented for the following train printers:

TP72	EBCDIC72 (1100 LPM)
TP96F	EBCDIC96 (1100 LPM)
TP64	EBCDIC64 (450/750 LPM)
TP64A	EBCDIC64A (450/750 LPM)
TP96S	EBCDIC96 (450/750 LPM)

SYSTEST - SCR/TP64

P1496 TP64 - "SCR" TESTS FOR TRAIN PRINTERS

SCR tests have been implemented for the following train printers:

TP72	EBCDIC72 (1100 LPM)
TP96F	EBCDIC96 (1100 LPM)
TP64	EBCDIC64 (450/750 LPM)
TP64A	EBCDIC64A (450/750 LPM)
TP96S	EBCDIC96 (450/750 LPM)

SYSTEST - SCR/TP64A

P1496 TP64A - "SCR" TESTS FOR TRAIN PRINTERS

SCR tests have been implemented for the following train printers:

TP72	EBCDIC72 (1100 LPM)
TP96F	EBCDIC96 (1100 LPM)
TP64	EBCDIC64 (450/750 LPM)
TP64A	EBCDIC64A (450/750 LPM)
TP96S	EBCDIC96 (450/750 LPM)

SYSTEST - SCR/TP96S

P1496 TP96S - "SCR" TESTS FOR TRAIN PRINTERS

SCR tests have been implemented for the following train printers:

TP72	EBCDIC72 (1100 LPM)
TP96F	EBCDIC96 (1100 LPM)
TP64	EBCDIC64 (450/750 LPM)
TP64A	EBCDIC64A (450/750 LPM)
TP96S	EBCDIC96 (450/750 LPM)

SYSTEST - UTIL/DKADDR

P1119 DKADDR - DISK PACK TYPE "235"

Printing of cylinder-head start addresses for disk pack type 235 has been added.

SYSTEST - UTIL/LOGMAPPER

P1115 LOGMAPPER - "SPO" INPUT ERROR MESSAGES

SPO input errors messages were written to the printer rather than the SPO. This has been corrected.

P1228 LOGMAPPER - "TODAY FOR LAST NN WEEKS"

The following changes have been made.

1. Allow AX message of "TODAY FOR LAST NN WEEKS" as well as "MM/DD/YY FOR LAST NN WEEKS".
2. SEG ARRAY on numeric date search has been corrected.
3. Fault on <month> <1 or >12 has been corrected.

SYSTEST - UTIL/PKTEST

P1669 PKTEST - RUN ON "BX385-235"

The system now also runs on BX385-235 disk packs. The program name has been changed from PKTEST225 to PKTEST.

SYSTEST - UTIL/PKTEST225

P1139 PKTEST225 - "B6800" COMPATIBLE "RD"

B6800 compatible result descriptors are now handled properly; i.e., word and character count instead of end memory address.

The program name has been changed from PKTEST225 to PKTEST.

P1140 PKTEST225 - "AX" "SPOTAB" FOR "TD830" DISPLAY

The SPO report message to the TD830 display has been corrected by adding ETX at the end of the screen to the message.

The program name has been changed from PKTEST225 to PKTEST.

SYSTEST - UTIL/RESHELPER

P1266 RESHELPER - ADD "TD830" DISPLAY AND "DPK235"

The system can now run with TD830 display and 235 disk packs.

SYSTEST - UTIL/LOGSTRIPPER

P1114 LOGSTRIPPER - FILE ADDED, DUPLICATE IDENTIFIER

The file LOCK MAINTHISTORY has been added.

The identifier MLLIBERR is defined in LOGSTRIPPER and also included in JOBFORMATTER, causing a compile error. This duplication has been corrected.

FTR ACTION TABLE

FTR	NOTE	SOFTWARE	PATCH	DESCRIPTION
008-0003	P1290	FILECOPY	30.0.0016	B6800 Vs. FILECOPY Heading
009-0010	P1615	INQ	30.0.0034	Loss of Last Column
009-0010	P1615	INQ	30.0.0037	Loss of Last Column
016-9009	P1650	COBOL	30.0.0077	OPEN EXTEND Leaving I/O Result
018-0007	D2586	MCP	30.0.0166	Test for WFL Data Decks Being
018-0007	D2586	WFL	30.0.	Test for WFL Data Decks Being
021-0008	P1508	FILECOPY	30.0.0021	Multiple Task Execution
042-0001	P1290	FILECOPY	30.0.0016	B6800 Vs. FILECOPY Heading
042-0007	P1667	MCP	30.0.0133	Update I/O Vs. IOERROR
042-0008	P1011	RJE	30.0.0009	Usercode Attributes for RJE St
042-0008	P1558	RJE	30.0.0037	WFL COMPILER with No PRIORITY
042-0008	P1011	RJE	30.0.0039	Usercode Attributes for RJE St
042-0008	P1558	RJE	30.0.0040	WFL COMPILER with No PRIORITY
042-0008	P1011	WFL	30.0.0018	Usercode Attributes for RJE St
042-0011	P1626	RJE	30.0.0044	Missing SKIP to Channel 1
049-0002	P1641	COBOL	30.0.0076	Value Clause
049-0004	P1626	RJE	30.0.0044	Missing SKIP to Channel 1
050-0001	P1580	LOGANALY	30.0.0023	LOG COMMENT Equals LOG OPERATO
050-0004	P1531	RJE	30.0.0035	Losing Blank Card at End of BI
050-0005	P1626	RJE	30.0.0044	Missing SKIP to Channel 1
050-0007	P1624	RJE	30.0.0042	BREAK on RJE RSC
050-0008	P1625	RJE	30.0.0043	FORMEDLP Array Vs. REMOTEPUNCH
052-0002	P1646	LOGGER	30.0.0021	INCLUDE, EXCLUDE Vs. Strings
060-7129	P1467	PLI	30.0.0015	"SET STATISTICS"
060-7147	P9073	ACR	30.0.0017	Delete Embedded Dataset Error
060-7150	P1063	REORG	30.0.0015	INVALID INDEX Reorganizing Bit
060-7152	D2267	CONTROLLER	30.0.	Four vs Three Digit System Ser
060-7152	D2267	FILECOPY	30.0.0007	Four vs Three Digit System Ser
060-7152	D2267	FILEDATA	30.0.0002	Four vs Three Digit System Ser
060-7152	D2267	LOGANALY	30.0.0006	Four vs Three Digit System Ser
060-7152	D2267	LOGGER	30.0.0005	Four vs Three Digit System Ser
095-0001	P1445	RECOVERY	30.0.0026	Stretch Header for Current Row
095-0005	P1559	MCP	30.0.0103	Volume Delete
095-0006	P1454	MCP	30.0.0077	READALABEL Update of Tape Kind
095-0007	P1121	RJE	30.0.0019	REBUILDPRINTQUEUE BD File Corr
095-0009	P1596	LOADDUMP	30.0.0005	Filler Dropped from COBOL File
106-1057	P1041	PLI	30.0.0005	Failure to Flag Large Precisio
106-1064	P1071	ACR	30.0.0050	DMSCLOSEERROR 2 at End of Halt
113-0871	P9101	ALGOL	30.0.0020	Real Valued Serial Numbers
113-0876	P9088	DASDL	30.0.0021	Minus Sign Problems
113-0878	P1640	BACKUP	30.0.0019	PB "<filename>" CP <unitno>
113-0889	P1521	ACR	30.0.0074	Recovery of Partitioned Struct
113-0889	P1521	RECOVERY	30.0.0028	Recovery of Partitioned Struct
113-0893	P1481	ALGOL	30.0.0071	THRU Loops in I/O Lists
114-1031	P1500	CANDE	30.0.0035	State Dump Problems
116-0107	P1056	NDL	30.0.0004	Increment Sequence Corrected
120-0012	P9101	ALGOL	30.0.0020	Real Valued Serial Numbers
120-0165	P1421	ESPOLINTRN	30.0.0023	BASIC, SETW Improvements
120-0170	D2249	MCP	30.0.	OF of GETSTATUS Waiting on <fi
120-0174	P9001	FILECOPY	30.0.0002	Added Task Exclude Index File
120-0176	P9093	REORG	30.0.0007	Invalid Reorganization of Empt
120-0188	P1365	FORTRAN	30.0.0020	INVALID OP in CHANGE , OPEN ,
120-0201	P1500	CANDE	30.0.0035	State Dump Problems
121-0205	P1481	ALGOL	30.0.0071	THRU Loops in I/O Lists
121-0206	P1659	FILEDATA	30.0.0012	TD830 Vs. Non-TD830 ODT
121-0206	P1659	MCP	30.0.0165	TD830 Vs. Non-TD830 ODT
122-5045	P1654	COBOL	30.0.0082	Sequence Dollar Option not Rea
125-0208	P1488	WFL	30.0.0047	Single Blank Character String
125-0211	P1531	RJE	30.0.0035	Losing Blank Card at End of BI
125-0214	D2586	MCP	30.0.0166	Test for WFL Data Decks Being
125-0214	D2586	WFL	30.0.	Test for WFL Data Decks Being
126-0176	P1475	JOBFORMAT	30.0.0008	Job Log I/O Errors
126-0184	P1506	FORTRAN	30.0.0028	Unexpected End of Initial Valu
126-0194	P1494	FORTRAN	30.0.0026	Formal Array in Implied DO Loo
126-0212	P1542	FORTRAN	30.0.0030	Source Programs in BCD Code
128-0271	P9068	FORTRAN	30.0.0004	OPT=1 Readlock Code
128-0368	P9068	FORTRAN	30.0.0004	OPT=1 Readlock Code
128-0389	P9024	MCP	30.0.	FIBLOCK and TIMESTAMP
128-0401	P9052	LOGANALY	30.0.0003	Spurious Records if File Equat
128-0437	P1010	RJE	30.0.0012	Blank Cards Inserted into Inpu
128-0442	P1103	ALGOL	30.0.0038	Error for Too Many Subscripts
128-0455	P1407	CONTROLLER	30.0.0007	PD Equal Vs. Nonresident Files
128-0459	P1365	FORTRAN	30.0.0020	INVALID OP in CHANGE , OPEN ,
128-0465	P1559	MCP	30.0.0103	Volume Delete
128-0466	P1457	CONTROLLER	30.0.0010	Filedata Catalog Resident Info
128-0466	P1457	MCP	30.0.0074	Filedata Catalog Resident Info
128-0472	P1553	MCP	30.0.0102	Controller Malfunction
128-0477	P1560	MCP	30.0.0104	AX
128-0483	P1011	RJE	30.0.0009	Usercode Attributes for RJE St
128-0483	P1558	RJE	30.0.0037	WFL COMPILER with No PRIORITY
128-0483	P1011	RJE	30.0.0039	Usercode Attributes for RJE St
128-0483	P1558	RJE	30.0.0040	WFL COMPILER with No PRIORITY

FTR ACTION TABLE

FTR	NOTE	SOFTWARE	PATCH	DESCRIPTION
128-0483	P1011	WFL	30.0.0018	Usercode Attributes for RJE St
130-0693	P9115	ESPOLINTRN	30.0.0003	REPLACE FOR NUMERIC Correction
130-0696	P9104	BASIC	30.0.0002	Erroneous Lines Written to ERR
130-0705	P9200	BASIC	30.0.0004	Syntax "IF END/MORE" Correctly
132-0099	P1323	MCP	30.0.0007	File Size Vs. Close with Crunc
132-0102	P9111	COBOL	30.0.0007	STRING get Erroneous Syntax Er
132-0110	P9199	BACKUP	30.0.0010	Clean Up Error Handling
140-0163	P9216	RJE	30.0.0007	Summary File Purge by *PB
140-0170	P1392	FILEDATA	30.0.0008	Running on B6800
140-0171	P1290	FILECOPY	30.0.0016	B6800 Vs. FILECOPY Heading
141-0252	P1050	WFL	30.0.0023	Handling of KIND Lists in OLD
141-0274	P1147	COBOL	30.0.0037	PERFORM Statements
141-0280	P1427	SCTABLEGEN	30.0.0003	Redundant Character Correction
141-0288	P1398	SORT	30.0.0001	MERGE Vs. UNITS=1
143-0225	D2225	FILECOPY	30.0.0003	EXCLUDE Syntax Correction
149-0359	D2443	BACKUP	30.0.0013	Disk and Pack Searched for Eac
149-0436	P9122	NDL	30.0.0003	EOF with Missing Define Crossh
149-0439	P9121	NDL	30.0.0002	One-Character Request Identifi
149-0440	D2240	NDL	30.0.	FINISH TRANSMIT with Delay
149-0483	P1479	MCP	30.0.0082	Protected Files Vs. Update Fil
149-0490	P1560	MCP	30.0.0104	AX
150-0062	P1365	FORTRAN	30.0.0020	INVALID OP in CHANGE , OPEN ,
151-0360	P1142	DCSTATUS	30.0.0002	TERMINAL MAXINPUT/MAXOUTPUT In
154-0157	P9112	COBOL	30.0.0010	Group Computational Data Items
154-0168	P1093	COBOL	30.0.0031	Xref Giving wrong Sequence Num
154-0169	P1093	COBOL	30.0.0031	Xref Giving wrong Sequence Num
156-0078	P1646	LOGGER	30.0.0021	INCLUDE, EXCLUDE Vs. Strings
162-0205	P9105	BINDER	30.0.0007	Large Host with Subroutines
166-0149	P9195	COBOL	30.0.0015	Include Size in Global Array I
166-0206	P1468	PLI	30.0.0016	Preprocessor INITIAL Attribute
166-0211	P9274	IN-OUTPUT	30.0.0023	CLOSE HERE on Empty Tape File
166-0214	P9274	IN-OUTPUT	30.0.0023	CLOSE HERE on Empty Tape File
166-0228	P9191	FILECOPY	30.0.0004	EXTRACTONEFILE vs. GETSTATUS A
166-0267	P1314	MCP	30.0.0003	CP Vs. HPT/PACK
166-0271	D2249	MCP	30.0.	OF of GETSTATUS Waiting on <fi
166-0272	P1289	FILECOPY	30.0.0015	FKINDS Array Eliminated
166-0278	P1503	CANDE	30.0.0039	Special Characters in UTILITY
166-0279	P1280	BDMSALGOL	30.0.0053	Binding DMSII Accesses
166-0279	P1280	BDMSPLI	30.0.0012	Binding DMSII Accesses
166-0291	P1554	MCP	30.0.0101	RCNTL
167-0130	P9126	PLINTRN	30.0.0004	ISAM , Delete First Record in
168-0285	P1432	MCP	30.0.0075	CP Vs. AIT
168-0292	P9133	WFL	30.0.0009	Eliminate INVALID INDEX on WFL
168-0364	P1551	LOADER	30.0.0008	LOADER Vs. Multiprocessor
169-0148	P9024	MCP	30.0.	FIBLOCK and TIMESTAMP
169-0157	P9118	LOGGER	30.0.0002	INCLUDE,EXCLUDE,CORRECTION Arr
169-0157	P9119	LOGGER	30.0.0003	EOF NO LABEL if Input Exceeds
170-0076	P9067	COBOL	30.0.0005	Constant Section Computational
170-0078	D2521	COBOL	30.0.0049	Floating Point Data Declaratio
170-0080	P1186	FILEDATA	30.0.0006	Input String >255 Characters
174-0138	P1092	COBOL	30.0.0030	Prevention of Integer Overflow
174-0149	P1198	PATCH	30.0.0003	Incorrect Sequence Error Issue
179-0243	P9000	FILECOPY	30.0.0001	Superfluous Comma in WFL Deck
179-0253	D2275	BACKUP	30.0.0011	Disallow "ND" When Used With "
180-0545	P1558	RJE	30.0.0037	WFL COMPILER with No PRIORITY
180-0545	P1558	RJE	30.0.0040	WFL COMPILER with No PRIORITY
180-0546	P1257	RJE	30.0.0021	Printing Jobs with More Than 1
181-0086	P9159	WFL	30.0.0012	Information Carry Over
181-0125	P9112	COBOL	30.0.0010	Group Computational Data Items
183-2901	P1682	ACR	30.0.0091	Insert/Remove Reset Bitvector
185-0238	P9252	DASDL	30.0.0028	NOT FOUND Exceptions if Entire
185-0245	P1335	IN-OUTPUT	30.0.0030	FILETYPE=4 Random I/O
188-0140	P9203	LOGANALY	30.0.0007	Log Date Correction
188-0164	P1681	ACR	30.0.0090	Embedded Data Set Paths Not In
192-0269	P1075	RJE	30.0.0017	Extra Page of Paper
192-0275	P9201	BASIC	30.0.0003	Equal Precedence for Division
192-0280	P1607	COBOL	30.0.0074	Report Writer
194-0205	P9076	UTIL	30.0.0010	No Reconstruction on Row Not D
194-0208	P9076	UTIL	30.0.0010	No Reconstruction on Row Not D
194-0211	P1072	UTIL	30.0.0019	File Listed Inside Parentheses
194-0220	P1554	MCP	30.0.0101	RCNTL
194-0225	P1622	UTIL	30.0.0050	Row Selection with Multiple Op
194-0226	D2586	MCP	30.0.0166	Test for WFL Data Decks Being
194-0226	D2586	WFL	30.0.	Test for WFL Data Decks Being
199-1103	D2586	MCP	30.0.0166	Test for WFL Data Decks Being
199-1103	D2586	WFL	30.0.	Test for WFL Data Decks Being
200-0203	P1182	MCP	30.0.	DBS D3 NOMEM
200-0203	P1182	PROPERTIES	30.0.0016	DBS D3 NOMEM
200-0217	P1512	MCP	30.0.0093	DMSOPEN Header
200-0237	P1683	ACR	30.0.0092	Attribute Error RSFILE.IOCLOCK
200-0240	P1390	ESPOL	30.0.0005	Bindinfo for Double Precision
201-0246	P1482	MCP	30.0.0086	SWAPPER Vs. EI

FTR ACTION TABLE

FTR	NOTE	SOFTWARE	PATCH	DESCRIPTION
201-0251	P1336	CONTROLLER	30.0.0002	MCS Message Causes SEG ARRAY
201-0255	P1500	CANDE	30.0.0035	State Dump Problems
202-0224	P1114	LOGSTRIPPER	30.0.0001	File Added, Duplicate Identifi
203-0137	P1071	ACR	30.0.0050	DMSCLOSEERROR 2 at End of Halt
203-0140	P1145	COBOL	30.0.0035	Recognition of Numeric Literal
203-0143	P1504	COBOL	30.0.0067	STATISTICS Option
208-7706	P1497	CANDE	30.0.0032	?QUIT Failure
208-7707	P1478	MCP	30.0.0080	Ending Banner Vs. Page Size
208-7711	P1354	RECOVERY	30.0.0024	Rebuild After Deleting Structu
209-0027	P9073	ACR	30.0.0017	Delete Embedded Dataset Error
210-0105	P8618	LOGGER	30.0.0001	UPDATE Option New File Creatio
210-0134	P1687	INQ	30.0.0038	Error on User Supplied Subscri
211-7741	P9200	BASIC	30.0.0004	Syntax "IF END/MORE" Correctly
215-0126	P1099	PLINTRN	30.0.0013	Character to Bit Conversion
222-0376	P1481	ALGOL	30.0.0071	THRU Loops in I/O Lists
226-0358	P1350	PTNCTL	30.0.0005	Wrong Family for Global Data
226-0364	P1583	RECOVERY	30.0.0030	ROLLBACK Using Tape Audit
226-0365	P1597	UTIL	30.0.0045	Tape Kind Set Correctly for Fi
226-0382	P1471	ACR	30.0.0071	Attribute Errors on Control Fi
226-0385	P1629	UTIL	30.0.0054	System Partitioned Files can b
226-0391	P1654	COBOL	30.0.0082	Sequence Dollar Option not Rea
227-0111	P1565	BACKUP	30.0.0015	Printer Selectable with Reel #
231-0383	P9026	LOGANALY	30.0.0001	Syntax Correction for SESSION
231-0395	P1282	COBOL	30.0.0053	ADD Statements
231-0405	P1659	FILEDATA	30.0.0012	TD830 Vs. Non-TD830 ODT
231-0405	P1659	MCP	30.0.0165	TD830 Vs. Non-TD830 ODT
231-0412	P1667	MCP	30.0.0133	Update I/O Vs. IOERROR
236-0028	P1051	WFL	30.0.0024	DATA or END Within File Names
236-0041	P1659	FILEDATA	30.0.0012	TD830 Vs. Non-TD830 ODT
236-0041	P1659	MCP	30.0.0165	TD830 Vs. Non-TD830 ODT
238-0160	P1257	RJE	30.0.0021	Printing Jobs with More Than 1
241-0096	P1498	CANDE	30.0.0033	Incorrect Destinations for ?RE
244-0150	P1090	COBOL	30.0.0028	RERUN Statement Branched Incor
244-0151	P1433	MCP	30.0.0076	CP Security
244-0152	P1432	MCP	30.0.0075	CP Vs. AIT
248-0033	P1482	MCP	30.0.0086	SWAPPER Vs. EI
248-0035	P1068	WFL	30.0.0027	Infinite Loop Following Securi
248-0101	P1150	COBOL	30.0.0040	Program Collating Sequence Cla
248-0105	P1495	SCR	30.0.0009	Allow Train Table Load
248-0107	P1148	COBOL	30.0.0038	COPY Statements
248-0108	P1284	BDMSCOBOL	30.0.0055	Data Base Section Syntax
248-0113	P1300	LOGGER	30.0.0010	Three vs. Four Digit System ID
249-0117	P9129	PLINTRN	30.0.0007	ISAM , Return Proper Value fro
249-0129	P1398	SORT	30.0.0001	MERGE Vs. UNITS=1
249-0131	P1492	DATACOM	30.0.0089	Prevent DCSTATUS Dump with Ext
249-0140	P1659	FILEDATA	30.0.0012	TD830 Vs. Non-TD830 ODT
249-0140	P1659	MCP	30.0.0165	TD830 Vs. Non-TD830 ODT
249-0143	P1659	FILEDATA	30.0.0012	TD830 Vs. Non-TD830 ODT
249-0143	P1659	MCP	30.0.0165	TD830 Vs. Non-TD830 ODT
252-0074	P1507	COBOL	30.0.0066	Syntax Group Comp Item Used as
252-0090	D2491	RJE	30.0.0024	LOGON Without PASSWORD on B700
254-0010	P9112	COBOL	30.0.0010	Group Computational Data Items
254-0017	P1659	FILEDATA	30.0.0012	TD830 Vs. Non-TD830 ODT
254-0017	P1659	MCP	30.0.0165	TD830 Vs. Non-TD830 ODT
255-2000	P9118	LOGGER	30.0.0002	INCLUDE,EXCLUDE,CORRECTION Arr
255-2013	P1646	LOGGER	30.0.0021	INCLUDE, EXCLUDE Vs. Strings
256-0101	P1006	COBOL	30.0.0021	INVALID INDEX on WAIT Statemen
256-0104	D2586	MCP	30.0.0166	Test for WFL Data Decks Being
256-0104	D2586	WFL	30.0.	Test for WFL Data Decks Being
261-0147	P8618	LOGGER	30.0.0001	UPDATE Option New File Creatio
261-0155	P9068	FORTRAN	30.0.0004	OPT=1 Readlock Code
261-0162	P9068	FORTRAN	30.0.0004	OPT=1 Readlock Code
261-0215	P1645	LOGGER	30.0.0020	PRIORITY Item Type INTEGER
261-0216	P1646	LOGGER	30.0.0021	INCLUDE, EXCLUDE Vs. Strings
261-0219	P1497	CANDE	30.0.0032	?QUIT Failure
261-0235	P1659	FILEDATA	30.0.0012	TD830 Vs. Non-TD830 ODT
261-0235	P1659	MCP	30.0.0165	TD830 Vs. Non-TD830 ODT
264-0055	P1106	CANDE	30.0.0014	Compile Time Task Attributes
264-0072	P9130	PLINTRN	30.0.0010	ISAM , Erroneous ISCLOSE Resul
264-0075	P1104	CANDE	30.0.0012	Empty Buffer Size in TERMINAL
264-0077	D2426	RJE	30.0.0016	Device Not Ready Abort QT of P
264-0082	P1659	FILEDATA	30.0.0012	TD830 Vs. Non-TD830 ODT
264-0082	P1659	MCP	30.0.0165	TD830 Vs. Non-TD830 ODT
264-0088	P1649	COBOL	30.0.0080	NEXT SENTENCE in INVALID KEY
265-0017	D2521	COBOL	30.0.0049	Floating Point Data Declaratio
272-0027	P1398	SORT	30.0.0001	MERGE Vs. UNITS=1
273-0019	P1010	RJE	30.0.0012	Blank Cards Inserted into Inpu
278-0030	P9130	PLINTRN	30.0.0010	ISAM , Erroneous ISCLOSE Resul
285-0030	P1689	INQ	30.0.0044	SEG ARRAY on Special Character
285-0031	P1691	INQ	30.0.0046	Nested Defines
286-0024	P1510	COBOL	30.0.0068	Global Direct Files Passed to
286-0027	P1542	FORTRAN	30.0.0030	Source Programs in BCD Code

FTR ACTION TABLE

FTR	NOTE	SOFTWARE	PATCH	DESCRIPTION
286-0037	P1659	FILEDATA	30.0.0012	TD830 Vs. Non-TD830 ODT
286-0037	P1659	MCP	30.0.0165	TD830 Vs. Non-TD830 ODT
286-0046	P1626	RJE	30.0.0044	Missing SKIP to Channel 1
290-0031	P1362	ESPOLINTRN	30.0.0020	BASIC, Making Arrays Present V
292-0107	P9127	PLINTRN	30.0.0005	ISAM , Numeric Keys
292-0120	P9205	LOGGER	30.0.0007	Include UNITNO Correction
292-0121	P9204	LOGGER	30.0.0008	Retention Report Item Correcti
292-0129	P1361	DCPPROGEN	30.0.0010	Auxiliary Logic Correction
292-0129	P1361	NDL	30.0.0005	Auxiliary Logic Correction
293-0067	P1541	FORTRAN	30.0.0029	Formals as "Implied-DO-Variabl
293-0073	P1121	RJE	30.0.0019	REBUILDPRINTQUEUE BD File Corr
293-0074	P1122	RJE	30.0.0020	INVALID INDEX with One Word Me
293-0077	P1397	MCP	30.0.0050	PB MT<number> Vs. ACMAX
293-0078	P1303	RJE	30.0.0029	REMOTECONTROL Vs. PRIMARYQUEUE
293-0079	P1500	CANDE	30.0.0035	State Dump Problems
293-0081	P1398	SORT	30.0.0001	MERGE Vs. UNITS=1
299-0113	P1413	COBOL	30.0.0047	SET Statements
299-0123	P9303	SOURCENDL	30.0.0001	Modifications to TD800 Request
299-0127	P1160	COBOL	30.0.0034	Listing of Omitted Source Lang
299-0128	P1146	COBOL	30.0.0036	Invalid Level Numbers
299-0129	P1043	COBOL	30.0.0023	Numeric Test Failed to Check S
299-0134	P1091	COBOL	30.0.0029	Code Listing
299-0141	P1151	COBOL	30.0.0042	Unnecessary Code Generation
299-0146	P1042	FILECOPY	30.0.	"(<usercode>) =" Syntax Error
299-0150	P9134	WFL	30.0.0010	Proper Information for Long CO
299-0167	P1365	FORTRAN	30.0.0020	INVALID OP in CHANGE , OPEN ,
300-7710	P1149	COBOL	30.0.0039	XREF
300-7727	P1115	LOGMAPPER	30.0.0001	SPO Input Error Messages
300-7740	P1433	MCP	30.0.0076	CP Security
300-7741	P1316	MCP	30.0.0005	CP File Security
301-0001	P1500	CANDE	30.0.0035	State Dump Problems
303-0001	D2573	FORTRAN	30.0.	Exponentiation Meaning
306-0012	P1489	WFL	30.0.0046	Correct HEX Function
306-0013	P1361	DCPPROGEN	30.0.0010	Auxiliary Logic Correction
306-0013	P1361	NDL	30.0.0005	Auxiliary Logic Correction
309-0010	P9134	WFL	30.0.0010	Proper Information for Long CO
309-0011	P9030	BACKUP	30.0.0001	BACKUP Vs. GETSTATUS
309-0021	P1646	LOGGER	30.0.0021	INCLUDE, EXCLUDE Vs. Strings
311-0166	P9024	MCP	30.0.	FIBLOCK and TIMESTAMP
311-0176	P1105	CANDE	30.0.0013	Nongraphic Characters in ?SS a
311-0185	P1561	MCP	30.0.0105	FORMATBUFFER
311-0187	P1408	MCP	30.0.0058	FAULTHANDLER Vs. Tagged Words
311-0193	P1499	CANDE	30.0.0034	Incorrect Error Messages
312-0032	P9110	BDMSCOBOL	30.0.0006	XREF DM Keys in SELECTION Exp
312-0036	P9019	UTIL	30.0.0008	Reconstruct Empty File
312-0039	P9194	COBOL	30.0.0017	Multiple COPY Statements
312-0050	P1581	UTIL	30.0.0044	Halt/Load Restart with Multipl
312-0058	P1594	ACR	30.0.0080	Deadlock During On-Line Recons
314-0010	P1391	MCP	30.0.0056	Direct Read from Schedule File
314-0011	P1159	COBOL	30.0.0033	TIME Intrinsic
314-0013	P1057	ACR	30.0.0053	Audited Bit Vector ZERO DISK A
314-0014	P9191	FILECOPY	30.0.0004	EXTRACTONEFILE vs. GETSTATUS A
319-0037	P8912	REORG	30.0.0001	Sort Error #4 on Internal Sort
319-0040	P9261	ACR	30.0.0045	Find Next Direct at Key > Valu
319-0041	P9096	ACR	30.0.0024	Incorrect Restart Record
319-0056	P1351	PRINTAUDIT	30.0.0005	Time Off by One Second
323-0020	P1107	CANDE	30.0.0015	Incorrect Statistics
324-0007	P1646	LOGGER	30.0.0021	INCLUDE, EXCLUDE Vs. Strings
326-0001	P9113	DUMPALL	30.0.0002	Packname Option for File Routi
329-0002	P8618	LOGGER	30.0.0001	UPDATE Option New File Creatio
331-0004	P1333	MCP	30.0.0028	PATHRES PBIT
331-0004	P1333	MCP	30.0.0040	PATHRES PBIT
331-0004	P1333	MCP	30.0.0063	PATHRES PBIT
335-0030	P9191	FILECOPY	30.0.0004	EXTRACTONEFILE vs. GETSTATUS A
337-0007	D2288	ACR	30.0.0044	Path Fixing Consistent
337-0007	D2288	PROPERTIES	30.0.0013	Path Fixing Consistent
337-0007	D2288	RECOVERY	30.0.0017	Path Fixing Consistent
339-0055	P1064	WFL	30.0.0026	Improve Handling End of File
341-0074	P1646	LOGGER	30.0.0021	INCLUDE, EXCLUDE Vs. Strings
343-0017	P1299	LOGGER	30.0.0009	INCLUDE, EXCLUDE Correction
343-0019	P1542	FORTRAN	30.0.0030	Source Programs in BCD Code
345-0815	D2424	ALGOL	30.0.0027	AUTOBIND and PPB
345-0815	D2424	BINDER	30.0.0011	AUTOBIND and PPB
345-0815	D2424	FORTRAN	30.0.0017	AUTOBIND and PPB
345-0903	P1638	ALGOL	30.0.0081	Scanning After XREFFILES
345-0917	P1295	LOGANALY	30.0.0009	STOPDATE Default
345-0918	P1296	LOGANALY	30.0.0010	MAINT Log Entries
345-0919	P1294	JOBFORMAT	30.0.0001	Cosmetic Changes
345-0919	P1294	LOGANALY	30.0.0011	Cosmetic Changes
345-0920	P1297	LOGANALY	30.0.0012	IOERRIRRCOUF Field
345-0921	P1298	LOGANALY	30.0.0013	INVALID INDEX in SORTIN Proced
347-0024	P9135	WFL	30.0.0011	Perform Job Rollout in More In

FTR ACTION TABLE

FTR	NOTE	SOFTWARE	PATCH	DESCRIPTION
348-0049	P1563	MCP	30.0.0109	IAD Vs. HLUNIT
348-0053	P1333	MCP	30.0.0028	PATHRES PBIT
348-0053	P1333	MCP	30.0.0040	PATHRES PBIT
348-0053	P1333	MCP	30.0.0063	PATHRES PBIT
348-0058	P1499	CANDE	30.0.0034	Incorrect Error Messages
348-0061	P1306	RJE	30.0.0027	RJELINKED Recovery Problem
348-0062	P1425	RJE	30.0.0034	Foreign Station Transfer
348-0066	P1497	CANDE	30.0.0032	?QUIT Failure
348-0068	P1493	MCP	30.0.0092	LIBMAINT Vs. TAPEPARITYRETRY
348-0068	P1493	MCP	30.0.0113	LIBMAINT Vs. TAPEPARITYRETRY
348-0069	P1493	MCP	30.0.0092	LIBMAINT Vs. TAPEPARITYRETRY
348-0069	P1493	MCP	30.0.0113	LIBMAINT Vs. TAPEPARITYRETRY
348-0071	P1304	RJE	30.0.0023	USER Option for WFL Secured Re
348-0072	P1304	RJE	30.0.0023	USER Option for WFL Secured Re
348-0073	P1558	RJE	30.0.0037	WFL COMPILER with No PRIORITY
348-0073	P1558	RJE	30.0.0040	WFL COMPILER with No PRIORITY
348-0078	D2560	RJE	30.0.0036	"*STATUS" Report NO TASKS ACTI
348-0089	P1566	RJE	30.0.0038	DESTNAME NEQ RJE Terminal Pr
348-0092	P1578	DIAGNOSTMCS	30.0.0002	Invalid Case Index in STASTATE
352-1016	P1649	COBOL	30.0.0080	NEXT SENTENCE in INVALID KEY
361-0071	P1120	BDMSPLI	30.0.0008	Reuse Scratch Stack Areas in K
361-0082	P1120	BDMSPLI	30.0.0008	Reuse Scratch Stack Areas in K
361-0084	P1302	PLI	30.0.0009	RESTARTED Task Attribute
361-0088	P1299	LOGGER	30.0.0009	INCLUDE, EXCLUDE Correction
361-0090	P1486	BDMSALGOL	30.0.0072	Allow "-" in Data Base Identif
361-0099	P1644	LOGGER	30.0.0019	REAL, INTEGER ITEMS Used as BR
363-0005	P9191	FILECOPY	30.0.0004	EXTRACTONEFILE vs. GETSTATUS A
366-0121	P1090	COBOL	30.0.0028	RERUN Statement Branched Incor
366-9015	P9026	LOGANALY	30.0.0001	Syntax Correction for SESSION
368-0039	P9177	ACR	30.0.0032	Initialization of Variable-For
368-0056	P1337	FILECOPY	30.0.0017	Card Input vs. UNITS=1
368-0063	P1634	RECOVERY	30.0.0036	Data Recovery Setting Bad EOF
369-0014	P9105	BINDER	30.0.0007	Large Host with Subroutines
369-0037	P1121	RJE	30.0.0019	REBUILDPRINTQUEUE BD File Corr
369-0047	P1680	ACR	30.0.0087	Result on SELECT Reject
369-0048	P1685	BUILDINQ	30.0.0009	Building Structures Not In Log
370-0009	P1398	SORT	30.0.0001	MERGE Vs. UNITS=1
372-0014	P9206	LOGGER	30.0.0006	BREAK ON PRIORITY
372-0200	P1398	SORT	30.0.0001	MERGE Vs. UNITS=1
373-0016	P1648	COBOL	30.0.0081	Illegal Syntax for READ not Ca
373-0022	P1649	COBOL	30.0.0080	NEXT SENTENCE in INVALID KEY
373-0033	P1360	DCPPROGEN	30.0.0011	Correct Address Used as a Byte
373-0037	P1492	DATACOM	30.0.0089	Prevent DCSTATUS Dump with Ext
374-0022	P9106	BINDER	30.0.0009	USE Cards Obeyed
374-0033	P1124	LTTABLEGEN	30.0.0002	User Specified Tables
374-0041	P9114	DUMPALL	30.0.0003	Correctly Print Variable Recor
381-0013	P1398	SORT	30.0.0001	MERGE Vs. UNITS=1
381-0022	P1353	RECOVERY	30.0.0023	Miscellaneous Rebuild Error
381-0025	P1635	UTIL	30.0.0055	PRINTIT Gives Error for Block
382-0010	P9126	PLINTRN	30.0.0004	ISAM , Delete First Record in
382-0020	P1500	CANDE	30.0.0035	State Dump Problems
383-0001	P9128	PLINTRN	30.0.0006	ISAM , Adding Records to Unblo
386-0017	P1646	LOGGER	30.0.0021	INCLUDE, EXCLUDE Vs. Strings
386-0018	D2586	MCP	30.0.0166	Test for WFL Data Decks Being
386-0018	D2586	WFL	30.0.	Test for WFL Data Decks Being
389-0032	D2274	RJE	30.0.0008	SAVE/READY Station Removal
389-0045	P9090	DASDL	30.0.0023	DIVIDE BY ZERO
389-0046	P9089	DASDL	30.0.0022	Sets with Name DATA
389-0047	P9251	ACR	30.0.0026	Too Many Table Levels Cause IN
389-0048	P1058	ACR	30.0.0052	SEG ARRAY Fault when Creating
390-0012	P1567	COBOL	30.0.0073	ANSI74 Default Write After Adv
390-0012	P1567	COBOL	30.0.0079	ANSI74 Default Write After Adv
393-0015	P1089	COBOL	30.0.0027	INVALID INDEX by LOCK Statemen
393-0028	P1240	COBOL	30.0.0048	XREF
393-0051	D2217	BACKUP	30.0.0002	"AX-<record #>" GTR Current Re
393-0053	P9178	ACR	30.0.0034	Standard Variable Format Avail
393-0060	P1411	COBOL	30.0.0045	MONITOR
393-0068	P1152	COBOL	30.0.0044	Undigit Literals in VALUE Clau
393-0069	P9178	ACR	30.0.0034	Standard Variable Format Avail
393-0082	P1147	COBOL	30.0.0037	PERFORM Statements
393-0083	P1088	COBOL	30.0.0026	Semicolon in NEXT SENTENCE and
393-0092	P1152	COBOL	30.0.0044	Undigit Literals in VALUE Clau
393-0111	P1614	RECOVERY	30.0.0031	Restart Reconstruction at Rest
393-0122	P1497	CANDE	30.0.0032	?QUIT Failure
393-9115	P1696	RECOVERY	30.0.0037	UNIDENTIFIED MISCELLANEOUS ERR
396-7002	P9112	COBOL	30.0.0010	Group Computational Data Items
397-0006	P1044	COBOL	30.0.0024	Incorrect Time Generated by AC
397-0015	P1346	ACR	30.0.0063	Abort Due to BAD DIVEST
399-9039	P9032	LOGANALY	30.0.0002	UA/UR Message Displays
402-0011	P1275	DASDL	30.0.0047	Improper Invalid Text
403-0006	P9175	BACKUP	30.0.0003	Remove Extra Page Skips
403-0013	P1543	FORTRAN	30.0.0031	Syntax For DO Statement

FTR ACTION TABLE

FTR	NOTE	SOFTWARE	PATCH	DESCRIPTION
405-0010	P9299	COBOL	30.0.0018	SEG ARRAY Termination of Compi
405-0011	P9030	BACKUP	30.0.0001	BACKUP Vs. GETSTATUS
405-0012	P9191	FILECOPY	30.0.0004	EXTRACTONEFILE vs. GETSTATUS A
405-0017	D2492	RJE	30.0.0026	Backup File Routing
405-0020	P1304	RJE	30.0.0023	USER Option for WFL Secured Re
405-0026	P1353	RECOVERY	30.0.0023	Miscellaneous Rebuild Error
405-0029	P1424	RJE	30.0.0031	SYSTEM/BACKUP Vs. Disk or Pack
405-0031	D2586	MCP	30.0.0166	Test for WFL Data Decks Being
405-0031	D2586	WFL	30.0.	Test for WFL Data Decks Being
405-0033	P1595	BUILDINQ	30.0.0008	Looping
405-0035	P1469	ACR	30.0.0070	All Zero Blocks/Records
405-0040	P1613	ACR	30.0.0082	Fix Audit Lastrecord; Zip COPY
406-0004	P9053	DASDL	30.0.0016	VERIFYSTORE Text for Remaps
406-0012	P9053	DASDL	30.0.0016	VERIFYSTORE Text for Remaps
406-0013	P9020	ACR	30.0.0014	INVALID INDEX, Variable Format
406-0015	P9289	FILECOPY	30.0.0011	Scanner Field Extensions
406-0017	P1651	DCALGOL	30.0.0082	Left Type Transfer Function
406-0029	P1586	COPYAUD-II	30.0.0004	Copy Primary Audit on Tape
406-0032	P1583	RECOVERY	30.0.0030	ROLLBACK Using Tape Audit
406-0033	P1631	RECOVERY	30.0.0035	ROLLBACK Gets Timestamp Mismat
406-0036	P1632	RECOVERY	30.0.0034	Erroneous Row Lockout
415-0025	P9013	ACR	30.0.0012	Return Correct Structure Numbe
415-0036	P1185	FILEDATA	30.0.0003	LFILES CAT
415-0078	P1139	PKTEST225	30.0.	B6800 Compatible RD
415-0083	P1141	DCSTATUS	30.0.0001	DCPANALYSIS Output Reformatted
415-0095	D2482	UTIL	30.0.0026	No File on CONTROLOLD
415-0101	P9185	ACR	30.0.0035	Elimination of Holes in Restar
415-0101	P9185	PRINTAUDIT	30.0.0003	Elimination of Holes in Restar
415-0101	P9185	PROPERTIES	30.0.0011	Elimination of Holes in Restar
415-0101	P9185	RECOVERY	30.0.0014	Elimination of Holes in Restar
415-0101	P9185	RECOVERY	30.0.0021	Elimination of Holes in Restar
415-0103	P1276	DMCTL	30.0.0017	Invalid Version Timestamp Mism
415-0107	P1274	ACR	30.0.0058	Missing HANDLEWRITEERROR Call
415-0110	P1355	UTIL	30.0.0028	Remove First Read Error Messag
415-0118	P1283	BDMSCOBOL	30.0.0054	Compares Involving Data Set Na
415-0133	P1353	RECOVERY	30.0.0023	Miscellaneous Rebuild Error
415-0138	P1447	DASDL	30.0.0052	Deleting Many Structures
415-0151	P1696	RECOVERY	30.0.0037	UNIDENTIFIED MISCELLANEOUS ERR
415-0157	P1695	LOADDUMP	30.0.0007	Invalid Syntax Error
415-0158	P1694	LOADDUMP	30.0.0006	Syntax Error When Mapping Bit
415-0160	P1613	ACR	30.0.0082	Fix Audit Lastrecord; Zip COPY
415-0164	P1678	ACR	30.0.0085	Errors in Recovery Close
415-0165	P1532	ACR	30.0.0078	UNIDENTIFIED MISCELLANEOUS ERR
415-0165	P1532	RECOVERY	30.0.0029	UNIDENTIFIED MISCELLANEOUS ERR
415-0169	P1679	ACR	30.0.0086	Deadlock Reconstruction Vs. Sy
417-2006	P1361	DCPPROGEN	30.0.0010	Auxiliary Logic Correction
417-2006	P1361	NDL	30.0.0005	Auxiliary Logic Correction
417-2009	P1500	CANDE	30.0.0035	State Dump Problems
417-2010	P1500	CANDE	30.0.0035	State Dump Problems
419-0007	P1300	LOGGER	30.0.0010	Three vs. Four Digit System ID
419-0010	P1493	MCP	30.0.0092	LIBMAINT Vs. TAPEPARITYRETRY
419-0010	P1493	MCP	30.0.0113	LIBMAINT Vs. TAPEPARITYRETRY
435-0027	P1508	FILECOPY	30.0.0021	Multiple Task Execution
435-0033	P1626	RJE	30.0.0044	Missing SKIP to Channel 1
437-0512	P1170	RSLOG	30.0.0002	Handle Multi-Record Log Entrie
437-0513	P1170	RSLOG	30.0.0002	Handle Multi-Record Log Entrie
437-0514	P1169	RSLOG	30.0.0003	Correct Date Printout
439-0002	P1088	COBOL	30.0.0026	Semicolon in NEXT SENTENCE and
439-0004	P1044	COBOL	30.0.0024	Incorrect Time Generated by AC
445-0013	P1587	ACR	30.0.0079	Serial Access of Invalid State
447-0012	P1187	DCPPROGEN	30.0.0009	Full Duplex Station Not Busy
450-0002	P1659	FILEDATA	30.0.0012	TD830 Vs. Non-TD830 ODT
450-0002	P1659	MCP	30.0.0165	TD830 Vs. Non-TD830 ODT
451-0002	P9191	FILECOPY	30.0.0004	EXTRACTONEFILE vs. GETSTATUS A
453-0001	P1508	FILECOPY	30.0.0021	Multiple Task Execution
454-9008	P1399	CONTROLLER	30.0.0005	DIR Vs. Input Scanner
455-0004	P1659	FILEDATA	30.0.0012	TD830 Vs. Non-TD830 ODT
455-0004	P1659	MCP	30.0.0165	TD830 Vs. Non-TD830 ODT
466-0002	P1482	MCP	30.0.0086	SWAPPER Vs. EI
467-0001	P1568	BACKUP	30.0.0017	Error for Colon Without ALGOL
549-0001	P1280	BDMSALGOL	30.0.0053	Binding DMSII Accesses
549-0001	P1280	BDMSPLI	30.0.0012	Binding DMSII Accesses

PATCH TABLE

SOFTWARE	PATCH	PRI	NOTE	DESCRIPTION
ACR	30.0.0002	32298	P8946	Interlock of Abort and Reconst
ACR	30.0.0005	32289	P8943	Correct Discontinuity Checking
ACR	30.0.0006	32279	D2219	Node Syntax
ACR	30.0.0008	32290	P8975	Handle Audit Block Size Change
ACR	30.0.0012	32285	P9013	Return Correct Structure Numbe
ACR	30.0.0013	34232	P9018	I/O Errors on Open
ACR	30.0.0014	32431	P9020	INVALID INDEX, Variable Format
ACR	30.0.0017	32428	P9073	Delete Embedded Dataset Error
ACR	30.0.0018	32424	D2277	SEGDESCABOVE
ACR	30.0.0019	32421	D2278	Display Structure Number
ACR	30.0.0020	32420	P9094	Fail When Print Statistics
ACR	30.0.0021	32438	D2279	Eliminate Warning if No Logica
ACR	30.0.0022	32418	D2244	Limit Error
ACR	30.0.0023	32713	P9084	Invalid DATAERROR 4
ACR	30.0.0024	32798	P9096	Incorrect Restart Record
ACR	30.0.0025	32716	P1017	ACCESSROUTINES Segmentation
ACR	30.0.0026	32717	P9251	Too Many Table Levels Cause IN
ACR	30.0.0027	32719	P1018	Improve I/O Error Display
ACR	30.0.0028	32795	P1019	Ordered Data Sets Out of Seque
ACR	30.0.0029	32725	P1020	Control File Locking
ACR	30.0.0030	32280	P9264	Eliminate MASKSEARCH in FIND K
ACR	30.0.0031	32791	P9181	Correct Ordered Set Paths
ACR	30.0.0032	32790	P9177	Initialization of Variable-For
ACR	30.0.0033	32791	P9181	Correct Ordered Set Paths
ACR	30.0.0034	32726	P9178	Standard Variable Format Avail
ACR	30.0.0035	32789	P9185	Elimination of Holes in Restar
ACR	30.0.0037	32783	P9256	Correct Path for Disjoint Orde
ACR	30.0.0038	32788	P9257	Data Corruption in Root Word
ACR	30.0.0039	32729	P9283	Warnings Suppressed
ACR	30.0.0040	32730	P9284	Compile-Time Printing
ACR	30.0.0043	32986	P9260	Accelerate Not Found Linear Se
ACR	30.0.0044	32781	D2288	Path Fixing Consistent
ACR	30.0.0045	32786	P9261	Find Next Direct at Key > Valu
ACR	30.0.0046	32984	P9281	Ordered Access Path
ACR	30.0.0047	32983	P9280	Partitioned Index Sets
ACR	30.0.0048	32720	D2291	Use of Description File
ACR	30.0.0049	32981	D2438	Buffer Core Improvement
ACR	30.0.0050	36969	P1071	DMSCLOSEERROR 2 at End of Halt
ACR	30.0.0051	32970	P1028	DASDL Update Timestamp
ACR	30.0.0052	32965	P1058	SEG ARRAY Fault when Creating
ACR	30.0.0053	32964	P1057	Audited Bit Vector ZERO DISK A
ACR	30.0.0054	33093	D2458	Version Overrides Recorded
ACR	30.0.0055	32960	P1199	Mark Stack DS if Abort Dies
ACR	30.0.0056	32959	P1200	SEG ARRAY if Pack Audit
ACR	30.0.0057	32958	P1153	Storage Close in Unaudited Dat
ACR	30.0.0058	33350	P1274	Missing HANDLEWRITEERROR Call
ACR	30.0.0059	31104	D2462	Read Ahead Buffering
ACR	30.0.0060	33430	D2497	New Method of Data Base Initia
ACR	30.0.0061	33342	P1344	SET ROWLOCK Read, Write Errors
ACR	30.0.0062	33343	P1345	Retry Disk Read on Checksum Er
ACR	30.0.0063	33339	P1346	Abort Due to BAD DIVEST
ACR	30.0.0064	33334	P1400	False Errorexit 9 in DIVEST Un
ACR	30.0.0065	33342	P1344	SET ROWLOCK Read, Write Errors
ACR	30.0.0066	31104	D2462	Read Ahead Buffering
ACR	30.0.0067	33342	P1344	SET ROWLOCK Read, Write Errors
ACR	30.0.0068	33332	P1342	Innocent Programs Hang on Dise
ACR	30.0.0070	30008	P1469	All Zero Blocks/Records
ACR	30.0.0071	33114	P1471	Attribute Errors on Control Fi
ACR	30.0.0072	31104	D2462	Read Ahead Buffering
ACR	30.0.0074	30011	P1521	Recovery of Partitioned Struct
ACR	30.0.0075	31104	D2462	Read Ahead Buffering
ACR	30.0.0076	31104	D2462	Read Ahead Buffering
ACR	30.0.0077	33117	P1533	ZERO FIND ADDRESS After INSERT
ACR	30.0.0078	32020	P1532	UNIDENTIFIED MISCELLANEOUS ERR
ACR	30.0.0079	32005	P1587	Serial Access of Invalid State
ACR	30.0.0080	31999	P1594	Deadlock During On-Line Recons
ACR	30.0.0081	32011	P1616	Error in SET TO ENDING Ordered
ACR	30.0.0082	33794	P1613	Fix Audit Lastrecord; Zip COPY
ACR	30.0.0085	33777	P1678	Errors in Recovery Close
ACR	30.0.0086	33781	P1679	Deadlock Reconstruction Vs. Sy
ACR	30.0.0087	34281	P1680	Result on SELECT Reject
ACR	30.0.0090	34286	P1681	Embedded Data Set Paths Not In
ACR	30.0.0091	34290	P1682	Insert/Remove Reset Bitvector
ACR	30.0.0092	34283	P1683	Attribute Error RSFILE.IOCLOCK
ACR	30.0.0093	34297	P1684	Halt Load Recovery Uses Wrong
ALGOL	30.0.0004	30070	D2427	String Implementation
ALGOL	30.0.0005	30070	D2427	String Implementation
ALGOL	30.0.0017	32449	P9072	$INCLUDE Interrupts $MAKEHOST
ALGOL	30.0.0018	32425	D2241	SEGDESCABOVE $ Option
ALGOL	30.0.0020	32668	P9101	Real Valued Serial Numbers
ALGOL	30.0.0021	32419	P9102	Compile-Time Display
ALGOL	30.0.0022	32685	D2265	Standardization of Compiler Fi

PATCH TABLE

SOFTWARE	PATCH	PRI	NOTE	DESCRIPTION
ALGOL	30.0.0024	32757	D2283	Update Info Level
ALGOL	30.0.0025	32801	D2256	$WARNSUPR Implemented
ALGOL	30.0.0026	32895	D2289	OFFSET Function
ALGOL	30.0.0027	32799	D2424	AUTOBIND and PPB
ALGOL	30.0.0029	32756	P9211	$LOADINFO
ALGOL	30.0.0032	32755	D2403	Warnings on Use of BCL Constru
ALGOL	30.0.0033	32755	D2403	Warnings on Use of BCL Constru
ALGOL	30.0.0034	32755	D2403	Warnings on Use of BCL Constru
ALGOL	30.0.0035	32590	D2425	XREF Procedure End Sequence Nu
ALGOL	30.0.0037	33018	D2442	Pointer in Truthset for Length
ALGOL	30.0.0038	33237	P1103	Error for Too Many Subscripts
ALGOL	30.0.0041	33769	D2484	DIRECTORYCONTROL, STACKSWAPPER
ALGOL	30.0.0042	33232	D2483	Uplevel Pointers
ALGOL	30.0.0043	33232	D2483	Uplevel Pointers
ALGOL	30.0.0044	33232	D2483	Uplevel Pointers
ALGOL	30.0.0045	33232	D2483	Uplevel Pointers
ALGOL	30.0.0046	33232	D2483	Uplevel Pointers
ALGOL	30.0.0047	33232	D2483	Uplevel Pointers
ALGOL	30.0.0048	33232	D2483	Uplevel Pointers
ALGOL	30.0.0049	33232	D2483	Uplevel Pointers
ALGOL	30.0.0050	33232	D2483	Uplevel Pointers
ALGOL	30.0.0055	33232	D2483	Uplevel Pointers
ALGOL	30.0.0071	32989	P1481	THRU Loops in I/O Lists
ALGOL	30.0.0073	32685	D2265	Standardization of Compiler Fi
ALGOL	30.0.0077	33934	P1605	Pointer Case Expressions
ALGOL	30.0.0080	33932	P1627	Critical Block Exit from PROCE
ALGOL	30.0.0081	34110	P1638	Scanning After XREFFILES
ALGOLTABLE	30.0.0004	32755	D2403	Warnings on Use of BCL Constru
ALGOLTABLE	30.0.0005	32755	D2403	Warnings on Use of BCL Constru
ALGOLTABLE	30.0.0007	33769	D2484	DIRECTORYCONTROL, STACKSWAPPER
ALGOLTABLE	30.0.0008	33232	D2483	Uplevel Pointers
ATTABLEGEN	30.0.0002	32078	P9285	Task Restructuring
ATTABLEGEN	30.0.0006	32700	D2440	NEWFILE Attribute
ATTABLEGEN	30.0.0008	33184	P1183	Floppy Disk
ATTABLEGEN	30.0.0010	33771	D2478	WFL Jobs See More Task Attribu
ATTABLEGEN	30.0.0012	33659	D2496	Swapper Enhancements
ATTABLEGEN	30.0.0013	33654	P1394	FILEKIND=FIRMWARE
ATTABLEGEN	30.0.0014	33618	D2508	CANDE Backup File Processing F
BACKUP	30.0.0001	31084	P9030	BACKUP Vs. GETSTATUS
BACKUP	30.0.0002	32333	D2217	"AX-<record #>" GTR Current Re
BACKUP	30.0.0003	32495	P9175	Remove Extra Page Skips
BACKUP	30.0.0007	32651	P9198	New Symbolic for III.0
BACKUP	30.0.0008	32657	P9176	SWAPPER QT's BACKUP
BACKUP	30.0.0009	32656	P1001	Execute BACKUP with RUN Statem
BACKUP	30.0.0010	32637	P9199	Clean Up Error Handling
BACKUP	30.0.0011	32635	D2275	Disallow "ND" When Used With "
BACKUP	30.0.0012	32596	D2421	RANGE Option Semantics
BACKUP	30.0.0013	32583	D2443	Disk and Pack Searched for Eac
BACKUP	30.0.0015	33806	P1565	Printer Selectable with Reel #
BACKUP	30.0.0016	33805	P1557	QT Malfunction
BACKUP	30.0.0017	33960	P1568	Error for Colon Without ALGOL
BACKUP	30.0.0018	34052	P1639	Unquoted String as Range Value
BACKUP	30.0.0019	34040	P1640	PB "<filename>" CP <unitno>
BASIC	30.0.0002	31869	P9104	Erroneous Lines Written to ERR
BASIC	30.0.0003	32745	P9201	Equal Precedence for Division
BASIC	30.0.0004	32930	P9200	Syntax "IF END/MORE" Correctly
BASIC	30.0.0006	32685	D2265	Standardization of Compiler Fi
BASIC	30.0.0007	32928	D2280	Scanin Replaced by TIMEINTRINS
BASIC	30.0.0011	32685	D2265	Standardization of Compiler Fi
BASIC	30.0.0012	33589	P1657	Syntax Error for Undeclared Im
BDMSALGOL	30.0.0015	32300	P8916	Expand Description Files
BDMSALGOL	30.0.0051	33074	D2488	OPEN TEMPORARY and CLOSE LOCK
BDMSALGOL	30.0.0052	33074	D2488	OPEN TEMPORARY and CLOSE LOCK
BDMSALGOL	30.0.0053	33070	P1280	Binding DMSII Accesses
BDMSALGOL	30.0.0072	33872	P1486	Allow "-" in Data Base Identif
BDMSCOBOL	30.0.0006	32541	P9110	XREF DM Keys in SELECTION Exp
BDMSCOBOL	30.0.0046	33302	P1412	Global Data Bases
BDMSCOBOL	30.0.0054	33321	P1283	Compares Involving Data Set Na
BDMSCOBOL	30.0.0055	33505	P1284	Data Base Section Syntax
BDMSCOBOL	30.0.0056	33074	D2488	OPEN TEMPORARY and CLOSE LOCK
BDMSCOBOL	30.0.0058	33074	D2488	OPEN TEMPORARY and CLOSE LOCK
BDMSCOBOL	30.0.0064	33074	D2488	OPEN TEMPORARY and CLOSE LOCK
BDMSPLI	30.0.0008	33069	P1120	Reuse Scratch Stack Areas in K
BDMSPLI	30.0.0010	33074	D2488	OPEN TEMPORARY and CLOSE LOCK
BDMSPLI	30.0.0011	33074	D2488	OPEN TEMPORARY and CLOSE LOCK
BDMSPLI	30.0.0012	33070	P1280	Binding DMSII Accesses
BINDER	30.0.0007	32684	P9105	Large Host with Subroutines
BINDER	30.0.0008	32685	D2265	Standardization of Compiler Fi
BINDER	30.0.0009	32686	P9106	USE Cards Obeyed
BINDER	30.0.0011	32799	D2424	AUTOBIND and PPB
BINDER	30.0.0025	33736	P1466	$STRICT Gives Syntax Errors
BUILDINQ	30.0.0001	32279	D2219	Node Syntax

PATCH TABLE

SOFTWARE	PATCH	PRI	NOTE	DESCRIPTION
BUILDINQ	30.0.0002	32300	P8916	Expand Description Files
BUILDINQ	30.0.0004	32957	P1201	INVALID INDEX
BUILDINQ	30.0.0005	32776	P1202	Missing Sets
BUILDINQ	30.0.0007	32024	P1535	BUILDINQ Fails With Logical Da
BUILDINQ	30.0.0008	33328	P1595	Looping
BUILDINQ	30.0.0009	34282	P1685	Building Structures Not In Log
BUILDREORG	30.0.0001	32300	P8916	Expand Description Files
BUILDREORG	30.0.0004	33120	P1534	INVALID INDEX With Too Many St
CANDE	30.0.0004	32685	D2265	Standardization of Compiler Fi
CANDE	30.0.0005	32815	P9107	Incorrect GRINDLIMIT Settings
CANDE	30.0.0006	32814	P9108	PRIVATE Security in MAKE Comma
CANDE	30.0.0008	32388	P1002	Store Length of FIND/REPLACE
CANDE	30.0.0010	32390	P9300	Eliminate HOLDQ Problems
CANDE	30.0.0011	32812	D2281	Mandatory Recoveryfile Filekin
CANDE	30.0.0012	31581	P1104	Empty Buffer Size in TERMINAL
CANDE	30.0.0013	31588	P1105	Nongraphic Characters in ?SS a
CANDE	30.0.0014	31587	P1106	Compile Time Task Attributes
CANDE	30.0.0015	31589	P1107	Incorrect Statistics
CANDE	30.0.0016	31586	D2444	MATCH Verb
CANDE	30.0.0017	32810	D2544	Listing Termination with BREAK
CANDE	30.0.0020	32807	P1234	Confusion with "?REPEAT" and "
CANDE	30.0.0021	32811	P1235	BEFORE Option in "?ENTER" Verb
CANDE	30.0.0024	33618	D2508	CANDE Backup File Processing F
CANDE	30.0.0025	33617	D2509	Parameters to START Jobs
CANDE	30.0.0029	33617	D2509	Parameters to START Jobs
CANDE	30.0.0030	33618	D2508	CANDE Backup File Processing F
CANDE	30.0.0031	33618	D2508	CANDE Backup File Processing F
CANDE	30.0.0032	33613	P1497	?QUIT Failure
CANDE	30.0.0033	33612	P1498	Incorrect Destinations for ?RE
CANDE	30.0.0034	33611	P1499	Incorrect Error Messages
CANDE	30.0.0035	33610	P1500	State Dump Problems
CANDE	30.0.0036	33608	P1501	REP, RET, TAKE, ENTER with CON
CANDE	30.0.0037	33609	P1502	Long File Equations with Compi
CANDE	30.0.0038	32685	D2265	Standardization of Compiler Fi
CANDE	30.0.0039	33607	P1503	Special Characters in UTILITY
CANDE	30.0.0040	33606	D2552	Userlimit for Schedule Session
CANDE	30.0.0041	33897	P1656	Lost Records with MOVE or INSE
CANDE	30.0.0042	33618	D2508	CANDE Backup File Processing F
CANDE	30.0.0043	33618	D2508	CANDE Backup File Processing F
COBOL	30.0.0005	32529	P9067	Constant Section Computational
COBOL	30.0.0007	32542	P9111	STRING get Erroneous Syntax Er
COBOL	30.0.0009	32539	P1003	Program Portability
COBOL	30.0.0010	32538	P9112	Group Computational Data Items
COBOL	30.0.0011	32537	D2284	SIGN Clause
COBOL	30.0.0012	32901	D2259	Decimal Accuracy in Compute St
COBOL	30.0.0014	32685	D2265	Standardization of Compiler Fi
COBOL	30.0.0015	32543	P9195	Include Size in Global Array I
COBOL	30.0.0016	32535	D2522	Statistics Dollar Option
COBOL	30.0.0017	32534	P9194	Multiple COPY Statements
COBOL	30.0.0018	32614	P9299	SEG ARRAY Termination of Compi
COBOL	30.0.0019	32533	P1004	Block Name for $STATISTICS Cod
COBOL	30.0.0020	32532	P1005	Reserved Word in Arithmetic St
COBOL	30.0.0021	32606	P1006	INVALID INDEX on WAIT Statemen
COBOL	30.0.0022	32531	P1007	Optimized PERFORM Statements
COBOL	30.0.0023	33039	P1043	Numeric Test Failed to Check S
COBOL	30.0.0024	33040	P1044	Incorrect Time Generated by AC
COBOL	30.0.0026	33042	P1088	Semicolon in NEXT SENTENCE and
COBOL	30.0.0027	32582	P1089	INVALID INDEX by LOCK Statemen
COBOL	30.0.0028	33049	P1090	RERUN Statement Branched Incor
COBOL	30.0.0029	33047	P1091	Code Listing
COBOL	30.0.0030	33043	P1092	Prevention of Integer Overflow
COBOL	30.0.0031	33044	P1093	Xref Giving wrong Sequence Num
COBOL	30.0.0032	33221	P1158	Generalized File Specifier in
COBOL	30.0.0033	33220	P1159	TIME Intrinsic
COBOL	30.0.0034	33219	P1160	Listing of Omitted Source Lang
COBOL	30.0.0035	33274	P1145	Recognition of Numeric Literal
COBOL	30.0.0036	33282	P1146	Invalid Level Numbers
COBOL	30.0.0037	33281	P1147	PERFORM Statements
COBOL	30.0.0038	33279	P1148	COPY Statements
COBOL	30.0.0039	33278	P1149	XREF
COBOL	30.0.0040	33277	P1150	Program Collating Sequence Cla
COBOL	30.0.0042	33275	P1151	Unnecessary Code Generation
COBOL	30.0.0044	33299	P1152	Undigit Literals in VALUE Clau
COBOL	30.0.0045	33300	P1411	MONITOR
COBOL	30.0.0047	33327	P1413	SET Statements
COBOL	30.0.0048	33326	P1240	XREF
COBOL	30.0.0049	33324	D2521	Floating Point Data Declaratio
COBOL	30.0.0052	33320	P1237	Size Error Code Overwritten
COBOL	30.0.0053	33322	P1282	ADD Statements
COBOL	30.0.0057	33498	P1285	Literals in Local Stack
COBOL	30.0.0059	33498	P1285	Literals in Local Stack
COBOL	30.0.0061	33502	D2500	Global Data Bases

PATCH TABLE

SOFTWARE	PATCH	PRI	NOTE	DESCRIPTION
COBOL	30.0.0063	33499	P1380	Condition Names
COBOL	30.0.0066	33676	P1507	Syntax Group Comp Item Used as
COBOL	30.0.0067	33677	P1504	STATISTICS Option
COBOL	30.0.0068	33915	P1510	Global Direct Files Passed to
COBOL	30.0.0069	32685	D2265	Standardization of Compiler Fi
COBOL	30.0.0070	33911	P1540	Index Data Names
COBOL	30.0.0071	33978	P1538	Set Dynamic File Attributes of
COBOL	30.0.0072	33979	P1539	Move Scaled Integer to Alphanu
COBOL	30.0.0073	33980	P1567	ANSI74 Default Write After Adv
COBOL	30.0.0074	34082	P1607	Report Writer
COBOL	30.0.0075	34090	P1604	Linage Lines Clause Left I/O R
COBOL	30.0.0076	34089	P1641	Value Clause
COBOL	30.0.0077	34158	P1650	OPEN EXTEND Leaving I/O Result
COBOL	30.0.0078	34155	P1655	ACCEPT Using Incorrect Qualifi
COBOL	30.0.0079	33980	P1567	ANSI74 Default Write After Adv
COBOL	30.0.0080	34154	P1649	NEXT SENTENCE in INVALID KEY
COBOL	30.0.0081	34153	P1648	Illegal Syntax for READ not Ca
COBOL	30.0.0082	34152	P1654	Sequence Dollar Option not Rea
CONTROLLER	30.0.0002	33126	P1336	MCS Message Causes SEG ARRAY
CONTROLLER	30.0.0003	33659	D2496	Swapper Enhancements
CONTROLLER	30.0.0004	33656	P1396	OL PK
CONTROLLER	30.0.0005	33643	P1399	DIR Vs. Input Scanner
CONTROLLER	30.0.0007	33620	P1407	PD Equal Vs. Nonresident Files
CONTROLLER	30.0.0010	33866	P1457	Filedata Catalog Resident Info
COPYAUD-II	30.0.0004	32006	P1586	Copy Primary Audit on Tape
CPU	30.0.0001	33665	P1377	Remove BCL Characters
CPU	30.0.0003	33572	P1703	OFFLINE Dollar Option
DASDL	30.0.0001	32300	P8916	Expand Description Files
DASDL	30.0.0002	32038	P8914	Reorganization of Data Set
DASDL	30.0.0003	32199	P8913	Invalid Audit AREASIZE with SE
DASDL	30.0.0008	32044	P8950	Variable Format Remaps
DASDL	30.0.0009	32279	D2219	Node Syntax
DASDL	30.0.0016	32441	P9053	VERIFYSTORE Text for Remaps
DASDL	30.0.0018	32427	P9074	Optimize Storage of Initial an
DASDL	30.0.0019	32422	P9095	Compile Reconstruct with DBNOD
DASDL	30.0.0021	32708	P9088	Minus Sign Problems
DASDL	30.0.0022	32707	P9089	Sets with Name DATA
DASDL	30.0.0023	32706	P9090	DIVIDE BY ZERO
DASDL	30.0.0024	32709	P9091	Non-Usercode Data Bases
DASDL	30.0.0025	32712	D2411	New Dollar Options
DASDL	30.0.0026	32718	P9262	Automatic Subsets of Sets
DASDL	30.0.0027	32720	D2291	Use of Description File
DASDL	30.0.0028	32722	P9252	NOT FOUND Exceptions if Entire
DASDL	30.0.0029	32723	P9263	Printing File Titles
DASDL	30.0.0030	32728	P9186	Group Keys For Random/Index Ra
DASDL	30.0.0032	32732	P9301	Control File Compatibility acr
DASDL	30.0.0034	32979	P1059	PARTITIONINFO in Compile Order
DASDL	30.0.0035	32970	P1028	DASDL Update Timestamp
DASDL	30.0.0036	32975	P1060	Global Data CHECKSUM
DASDL	30.0.0037	32976	D2437	REORGANIZE GLOBAL DATA Extende
DASDL	30.0.0038	33094	P1203	Group Keys and REORGANIZE
DASDL	30.0.0039	33095	P1204	UPDATE Change of Data Set Subt
DASDL	30.0.0040	33098	P1205	Error Recovery Improvement
DASDL	30.0.0041	33097	P1206	Compiler Fault
DASDL	30.0.0042	33099	P1207	Machine ID in Heading
DASDL	30.0.0043	33096	D2460	Maintain UPDATE Level
DASDL	30.0.0044	33102	D2461	User Defined Dollar Options
DASDL	30.0.0045	33103	P1208	Delete Varying Items
DASDL	30.0.0046	31104	D2462	Read Ahead Buffering
DASDL	30.0.0047	33428	P1275	Improper Invalid Text
DASDL	30.0.0049	33430	D2497	New Method of Data Base Initia
DASDL	30.0.0050	33107	D2534	UPDATE and INITIALIZE
DASDL	30.0.0052	33109	P1447	Deleting Many Structures
DASDL	30.0.0053	33110	P1446	Numeric Item Size of Zero
DASDL	30.0.0054	33111	P1472	Label Source Input Correctly
DASDL	30.0.0055	33112	P1473	NEWTAPE File in Error
DASDL	30.0.0056	33113	P1474	UNIQUE Compiles Correctly
DASDL	30.0.0057	33125	P1526	INCLUDE Dollar Option
DASDL	30.0.0058	33115	P1527	Default Initial Value
DASDL	30.0.0059	33121	P1528	INITIALVALUE in DEFAULT Statem
DASDL	30.0.0060	33122	P1529	Property Buffer Declaration
DASDL	30.0.0061	33116	P1530	Build CONTROLITEMNODE Correctl
DASDL	30.0.0064	32016	P1584	Spurious Error During DASDL Up
DASDL	30.0.0065	32014	P1588	Prevent Loop on Dollar Card
DASDL	30.0.0067	32010	P1617	Free Property Buffer for Acces
DASDL	30.0.0068	33780	P1686	Prevent Spurious Message
DATACOM	30.0.0089	32561	P1492	Prevent DCSTATUS Dump with Ext
DATACOM	30.0.0090	32560	D2545	DCRECON No Longer Visible
DCALGOL	30.0.0082	34278	P1651	Left Type Transfer Function
DCPPROGEN	30.0.0002	32394	P8993	DCP Fault Report
DCPPROGEN	30.0.0004	32395	P8999	INITIALIZE RETRY With LOCAL TA
DCPPROGEN	30.0.0005	32397	P9050	TERMINATE BLOCK

PATCH TABLE

SOFTWARE	PATCH	PRI	NOTE	DESCRIPTION
DCPPROGEN	30.0.0006	33003	P9266	Correct Fetch Stored Text
DCPPROGEN	30.0.0007	33004	P1009	Improved Switched Line Handlin
DCPPROGEN	30.0.0008	33006	P1143	LINE INTERROGATE
DCPPROGEN	30.0.0009	33007	P1187	Full Duplex Station Not Busy
DCPPROGEN	30.0.0010	32762	P1361	Auxiliary Logic Correction
DCPPROGEN	30.0.0011	32763	P1360	Correct Address Used as a Byte
DCPTESTGEN	30.0.0001	33005	P1094	$NEWSEQERR Option
DCPTESTGEN	30.0.0002	33225	P1108	Require Good Parity on Disk
DCSTATUS	30.0.0001	32760	P1141	DCPANALYSIS Output Reformatted
DCSTATUS	30.0.0002	32761	P1142	TERMINAL MAXINPUT/MAXOUTPUT In
DFO	30.0.0001	33665	P1377	Remove BCL Characters
DFOSUTEST	30.0.0001	33665	P1377	Remove BCL Characters
DIAGNOSTMCS	30.0.0002	34080	P1578	Invalid Case Index in STASTATE
DISKEX	30.0.0001	33665	P1377	Remove BCL Characters
DKADDR	30.0.0001	33241	P1119	Disk Pack Type 235
DMALGOL	30.0.0001	32279	D2219	Node Syntax
DMALGOL	30.0.0019	32426	D2276	Print D1 Stacksize
DMALGOL	30.0.0078	33933	P1606	Data Pools in I/O Lists
DMCTL	30.0.0001	32279	D2219	Node Syntax
DMCTL	30.0.0004	32300	P8916	Expand Description Files
DMCTL	30.0.0006	32423	P9082	Optional File When Initializin
DMCTL	30.0.0007	32739	P9092	Optional Programdump for Contr
DMCTL	30.0.0008	32715	P1021	Better Displays for Control Fi
DMCTL	30.0.0009	32724	P1022	Invalid Control File Creation
DMCTL	30.0.0011	32732	P9301	Control File Compatibility acr
DMCTL	30.0.0012	32970	P1028	DASDL Update Timestamp
DMCTL	30.0.0013	32977	D2436	Halt/Load Bit Override
DMCTL	30.0.0014	33092	D2435	Unaudited Data Base Control Fi
DMCTL	30.0.0015	33093	D2458	Version Overrides Recorded
DMCTL	30.0.0016	33096	D2460	Maintain UPDATE Level
DMCTL	30.0.0017	33105	P1276	Invalid Version Timestamp Mism
DMCTL	30.0.0018	33430	D2497	New Method of Data Base Initia
DMCTL	30.0.0019	33337	P1347	Separate Include of CF, CFOLD
DMCTL	30.0.0020	33336	P1349	Check for Valid Structure Numb
DMCTL	30.0.0022	33783	P1636	Rebuild from II.8 Dump Tapes
DSK	30.0.0001	33228	P1117	Title Change
DSK	30.0.0002	33665	P1377	Remove BCL Characters
DUMPALL	30.0.0001	32506	P9031	Suppress Extra Lines in LAN Li
DUMPALL	30.0.0002	32500	P9113	Packname Option for File Routi
DUMPALL	30.0.0003	32496	P9114	Correctly Print Variable Recor
DUMPALL	30.0.0004	32222	P9213	Correct Pack Handling
DUMPALL	30.0.0005	32634	P9214	Propagate Save Factor
DUMPALL	30.0.0006	32633	P9215	Revise Tape Handling in DMPMT
DUMPANALY	30.0.0004	33635	P1233	Split DBS into Active/Nonactiv
ESPOL	30.0.0001	32685	D2265	Standardization of Compiler Fi
ESPOL	30.0.0002	32685	D2265	Standardization of Compiler Fi
ESPOL	30.0.0005	33770	P1390	Bindinfo for Double Precision
ESPOLINTRN	30.0.0003	32752	P9115	REPLACE FOR NUMERIC Correction
ESPOLINTRN	30.0.0004	32523	P9116	ATTRIBSEARCHER Checking Mnemon
ESPOLINTRN	30.0.0007	32485	D2402	Lower Bounds for FORTRAN Array
ESPOLINTRN	30.0.0011	33222	P1156	TAN and COTAN Intrinsics
ESPOLINTRN	30.0.0012	33223	P1157	DGAMMA Intrinsic
ESPOLINTRN	30.0.0020	33287	P1362	BASIC, Making Arrays Present V
ESPOLINTRN	30.0.0023	33594	P1421	BASIC, SETW Improvements
ESPOLINTRN	30.0.0024	32990	P1420	ALGOL Output Array Element
ESPSIM	30.0.0001	32917	P1165	Handle CANT-READ Correctly
FILECOPY	30.0.0001	34336	P9000	Superfluous Comma in WFL Deck
FILECOPY	30.0.0002	32335	P9001	Added Task Exclude Index File
FILECOPY	30.0.0003	31090	D2225	EXCLUDE Syntax Correction
FILECOPY	30.0.0004	32681	P9191	EXTRACTONEFILE vs. GETSTATUS A
FILECOPY	30.0.0006	32642	D2258	NO FILE on INCLUDE statement
FILECOPY	30.0.0007	32636	D2267	Four vs Three Digit System Ser
FILECOPY	30.0.0008	32619	P9189	Expired Task
FILECOPY	30.0.0009	32617	D2272	Savefactor Zero vs FILECOPY
FILECOPY	30.0.0010	32620	P9192	Summary Output Reformat
FILECOPY	30.0.0011	32610	P9289	Scanner Field Extensions
FILECOPY	30.0.0012	32609	P9290	AUXFILE and AUXBUF Field Exten
FILECOPY	30.0.0013	33173	P1291	Large INCLUDE and EXCLUDE LIST
FILECOPY	30.0.0014	33166	D2485	SYNTAX Modifier
FILECOPY	30.0.0015	33138	P1289	FKINDS Array Eliminated
FILECOPY	30.0.0016	33136	P1290	B6800 Vs. FILECOPY Heading
FILECOPY	30.0.0017	33130	P1337	Card Input vs. UNITS=1
FILECOPY	30.0.0020	33837	P1477	Loop on Input
FILECOPY	30.0.0021	33864	P1508	Multiple Task Execution
FILEDATA	30.0.0002	32636	D2267	Four vs Three Digit System Ser
FILEDATA	30.0.0003	33190	P1185	LFILES CAT
FILEDATA	30.0.0006	33181	P1186	Input String >255 Characters
FILEDATA	30.0.0008	33172	P1392	Running on B6800
FILEDATA	30.0.0009	33133	P1339	Determine System Type
FILEDATA	30.0.0012	34261	P1659	TD830 Vs. Non-TD830 ODT
FORTRAN	30.0.0004	32486	P9068	OPT=1 Readlock Code
FORTRAN	30.0.0005	32685	D2265	Standardization of Compiler Fi

PATCH TABLE

SOFTWARE	PATCH	PRI	NOTE	DESCRIPTION
FORTRAN	30.0.0007	32482	P9292	$ Option LEVEL=N, N>15
FORTRAN	30.0.0008	32481	P9293	$INCLUDE of a Null Range
FORTRAN	30.0.0009	32480	P9294	Improvements to Trailer Inform
FORTRAN	30.0.0010	32479	P9295	INVALID INDEX on Oversized Com
FORTRAN	30.0.0011	32478	P9296	Use of EXIT as a Subroutine
FORTRAN	30.0.0013	32755	D2403	Warnings on Use of BCL Constru
FORTRAN	30.0.0014	32485	D2402	Lower Bounds for FORTRAN Array
FORTRAN	30.0.0017	32799	D2424	AUTOBIND and PPB
FORTRAN	30.0.0018	32998	D2446	VECTORMODEISALLOWED Deimplemen
FORTRAN	30.0.0020	32996	P1365	INVALID OP in CHANGE , OPEN ,
FORTRAN	30.0.0021	32994	P1367	Correction to VECTORMODEISALLO
FORTRAN	30.0.0022	32995	D2501	Installation Intrinsics in FOR
FORTRAN	30.0.0023	32995	D2501	Installation Intrinsics in FOR
FORTRAN	30.0.0026	32988	P1494	Formal Array in Implied DO Loo
FORTRAN	30.0.0027	32685	D2265	Standardization of Compiler Fi
FORTRAN	30.0.0028	33924	P1506	Unexpected End of Initial Valu
FORTRAN	30.0.0029	33923	P1541	Formals as "Implied-DO-Variabl
FORTRAN	30.0.0030	33922	P1542	Source Programs in BCD Code
FORTRAN	30.0.0031	33921	P1543	Syntax For DO Statement
HELPINQ	30.0.0001	32269	P9087	Correct Syntax Diagram
HELPINQ	30.0.0004	32777	D2410	Report Facility for DMSII INQU
HELPINQ	30.0.0005	32778	D2409	Multiple-Statement Defines
HELPINQ	30.0.0006	32780	D2408	Printer File Attributes
HELPINQ	30.0.0007	32962	D2459	New HELP File
HELPINQ	30.0.0009	32777	D2410	Report Facility for DMSII INQU
IN-OUTPUT	30.0.0011	30481	P1325	FILETYPE=6 RECORDCOUNT
IN-OUTPUT	30.0.0023	30478	P9274	CLOSE HERE on Empty Tape File
IN-OUTPUT	30.0.0024	30477	P9273	PROGRAMDUMP, New IOCB
IN-OUTPUT	30.0.0030	30485	P1335	FILETYPE=4 Random I/O
IN-OUTPUT	30.0.0037	31591	D2498	Backup File Features
IN-OUTPUT	30.0.0042	30486	P1382	CARRIAGECONTROL=CTLASA or CTL3
IN-OUTPUT	30.0.0047	30475	D2287	I/O Subsystem Improvements
IN-OUTPUT	30.0.0048	30488	D2516	BCL Backup Files
IN-OUTPUT	30.0.0061	30488	D2516	BCL Backup Files
IN-OUTPUT	30.0.0062	30491	P1410	Status Change Vs. RSVP Replies
IN-OUTPUT	30.0.0066	31591	D2498	Backup File Features
IN-OUTPUT	30.0.0068	30491	P1410	Status Change Vs. RSVP Replies
IN-OUTPUT	30.0.0073	30491	P1410	Status Change Vs. RSVP Replies
IN-OUTPUT	30.0.0114	33962	P1574	LEB Resize Missed by OUTPUTREE
INQ	30.0.0001	32279	D2219	Node Syntax
INQ	30.0.0002	32270	P9015	SEG ARRAY Error
INQ	30.0.0003	32434	P9016	INVALID INDEX
INQ	30.0.0005	32300	P8916	Expand Description Files
INQ	30.0.0007	32429	P9075	INVALID INDEX at Initializatio
INQ	30.0.0008	31506	P9086	Correct Syntax Error
INQ	30.0.0009	32268	P9085	Option QUOTES
INQ	30.0.0010	32797	D2251	QUIT
INQ	30.0.0011	32796	P1023	Correct Check for Name Conflic
INQ	30.0.0012	32792	P9179	Display of Exception Value
INQ	30.0.0016	32779	P9259	SAVE/RESTORE Large Texts
INQ	30.0.0018	32777	D2410	Report Facility for DMSII INQU
INQ	30.0.0019	32778	D2409	Multiple-Statement Defines
INQ	30.0.0020	32780	D2408	Printer File Attributes
INQ	30.0.0021	32966	P1062	Improved Search Algorithms
INQ	30.0.0022	32963	P1061	Large Selection Conditions
INQ	30.0.0027	30014	P1525	Qualification Resolution
INQ	30.0.0028	32019	P1536	INQUIRY Uses Subsets Improperl
INQ	30.0.0029	31996	D2581	Right-Justify Numerics in DISP
INQ	30.0.0030	31998	P1619	Correct Report Heading Feature
INQ	30.0.0031	31996	D2581	Right-Justify Numerics in DISP
INQ	30.0.0032	30013	P1524	SEG ARRAY Fault on Input
INQ	30.0.0033	32000	P1618	GENERATE in a Parametric Defin
INQ	30.0.0034	33801	P1615	Loss of Last Column
INQ	30.0.0035	33796	P1620	Report Page Eject
INQ	30.0.0037	33801	P1615	Loss of Last Column
INQ	30.0.0038	33788	P1687	Error on User Supplied Subscri
INQ	30.0.0039	33775	P1688	Double Space on CRT Terminals
INQ	30.0.0040	33774	P1623	Range Test
INQ	30.0.0044	33753	P1689	SEG ARRAY on Special Character
INQ	30.0.0045	33754	P1690	RESTORE Large Text
INQ	30.0.0046	33751	P1691	Nested Defines
INQ	30.0.0048	32777	D2410	Report Facility for DMSII INQU
INQ	30.0.0054	34289	D2587	Error Messages
INQ	30.0.0055	34292	P1692	Correct Abbreviations, Report
INQ	30.0.0056	34299	P1693	Software Error
INTERFACE	30.0.0004	32300	P8916	Expand Description Files
INTERFACE	30.0.0009	32970	P1028	DASDL Update Timestamp
IOIOP	30.0.0001	33229	P1118	Accelerate Data Compare
IXREF	30.0.0001	30070	D2427	String Implementation
IXREF	30.0.0002	30070	D2427	String Implementation
IXREF	30.0.0004	33700	D2526	Change INTERACTIVEXREF File Ti
IXREF	30.0.0005	33699	D2527	Versions for INTERACTIVEXREF F

PATCH TABLE

SOFTWARE	PATCH	PRI	NOTE	DESCRIPTION
IXREF	30.0.0006	33742	D2525	Add PRINTER Option for HELP Co
IXREF	30.0.0007	33741	P1430	Modify Output Format
IXREF	30.0.0008	33740	P1431	Correctly Expand Defines
JOBFORMAT	30.0.0001	33147	P1294	Cosmetic Changes
JOBFORMAT	30.0.0002	33131	D2486	PE9MT Vs. PETAPE
JOBFORMAT	30.0.0003	33549	P1293	Swapjob Number and Job Name
JOBFORMAT	30.0.0005	33638	P1383	Job Scheduletime
JOBFORMAT	30.0.0007	33867	P1422	Checkbit Result Mapping
JOBFORMAT	30.0.0008	33853	P1475	Job Log I/O Errors
LCOBOL	30.0.0006	33231	D2417	FILEKIND for LCOBOL Symbol, Co
LCOBOL	30.0.0008	33514	P1368	Correct Bad Literals, OPTIM
LCOBOL	30.0.0010	33919	P1608	Number of Characters in a Grou
LCOBOL	30.0.0011	33918	P1593	Patching LCOBOL Source Program
LOADDUMP	30.0.0001	32300	P8916	Expand Description Files
LOADDUMP	30.0.0005	32012	P1596	Filler Dropped from COBOL File
LOADDUMP	30.0.0006	32009	P1694	Syntax Error When Mapping Bit
LOADDUMP	30.0.0007	33787	P1695	Invalid Syntax Error
LOADER	30.0.0002	33626	P1343	Change Panel, SPO Display
LOADER	30.0.0003	32866	D2273	IV vs 235
LOADER	30.0.0005	33644	P1455	Disk Pack I/O Errors
LOADER	30.0.0006	33855	P1460	Correct EOJ Message
LOADER	30.0.0008	33813	P1551	LOADER Vs. Multiprocessor
LOGANALY	30.0.0001	31089	P9026	Syntax Correction for SESSION
LOGANALY	30.0.0002	31088	P9032	UA/UR Message Displays
LOGANALY	30.0.0003	32505	P9052	Spurious Records if File Equat
LOGANALY	30.0.0004	33819	D2554	B6800 CPUERROR
LOGANALY	30.0.0006	32636	D2267	Four vs Three Digit System Ser
LOGANALY	30.0.0007	32624	P9203	Log Date Correction
LOGANALY	30.0.0008	33193	P1133	SO and RO Options Display
LOGANALY	30.0.0009	33164	P1295	STOPDATE Default
LOGANALY	30.0.0010	33163	P1296	MAINT Log Entries
LOGANALY	30.0.0011	33147	P1294	Cosmetic Changes
LOGANALY	30.0.0012	33146	P1297	IOERRIRRCOUF Field
LOGANALY	30.0.0013	33145	P1298	INVALID INDEX in SORTIN Proced
LOGANALY	30.0.0014	33819	D2554	B6800 CPUERROR
LOGANALY	30.0.0017	33819	D2554	B6800 CPUERROR
LOGANALY	30.0.0020	33827	P1573	132 Character Output
LOGANALY	30.0.0021	33819	D2554	B6800 CPUERROR
LOGANALY	30.0.0022	33968	D2561	PROCESSOR ID
LOGANALY	30.0.0023	33952	P1580	LOG COMMENT Equals LOG OPERATO
LOGANALY	30.0.0024	34273	P1637	LOG MAINT Correction
LOGGER	30.0.0001	31351	P8618	UPDATE Option New File Creatio
LOGGER	30.0.0002	32673	P9118	INCLUDE,EXCLUDE,CORRECTION Arr
LOGGER	30.0.0003	32674	P9119	EOF NO LABEL if Input Exceeds
LOGGER	30.0.0005	32636	D2267	Four vs Three Digit System Ser
LOGGER	30.0.0006	32629	P9206	BREAK ON PRIORITY
LOGGER	30.0.0007	32626	P9205	Include UNITNO Correction
LOGGER	30.0.0008	32625	P9204	Retention Report Item Correcti
LOGGER	30.0.0009	33161	P1299	INCLUDE, EXCLUDE Correction
LOGGER	30.0.0010	33144	P1300	Three vs. Four Digit System ID
LOGGER	30.0.0014	33861	D2536	JOBSUMMARY File Record Size In
LOGGER	30.0.0015	33841	D2537	PRE27 Option Removed
LOGGER	30.0.0016	33850	D2538	JOBENTRYTIME , JOBQUEUEDTIME
LOGGER	30.0.0017	34047	P1642	Version Heading Changed
LOGGER	30.0.0018	34046	P1643	Log Title Display Truncated
LOGGER	30.0.0019	34038	P1644	REAL, INTEGER ITEMS Used as BR
LOGGER	30.0.0020	34037	P1645	PRIORITY Item Type INTEGER
LOGGER	30.0.0021	34035	P1646	INCLUDE, EXCLUDE Vs. Strings
LOGMAPPER	30.0.0001	33242	P1115	SPO Input Error Messages
LOGMAPPER	30.0.0002	33480	P1228	TODAY FOR LAST NN WEEKS
LOGSTRIPPER	30.0.0001	33230	P1114	File Added, Duplicate Identifi
LTTABLEGEN	30.0.0001	33201	P1130	450/750 LPM Train Printers
LTTABLEGEN	30.0.0002	33200	P1124	User Specified Tables
MAINTMCS	30.0.0001	33561	P1271	Improper Test Number Selection
MAINTMCS	30.0.0002	33562	D2479	Datacom-to-Disk Syntax
MAINTMCS	30.0.0004	33564	D2480	CANCEL Command for DCPTEST
MAINTMCS	30.0.0005	34079	D2582	Front-End-Control Selection
MCP	30.0.0002	33272	P1313	LINKLLISTIT
MCP	30.0.0003	33270	P1314	CP Vs. HPT/PACK
MCP	30.0.0004	32690	P9286	DM6700 Vestiges Removed
MCP	30.0.0005	33154	P1316	CP File Security
MCP	30.0.0006	33157	P1317	Locks Vs. PAST
MCP	30.0.0007	33129	P1323	File Size Vs. Close with Crunc
MCP	30.0.0010	33546	P1324	Make WFL Use Direct Array
MCP	30.0.0012	32690	P9286	DM6700 Vestiges Removed
MCP	30.0.0013	33544	P1326	INFODUMPER Option
MCP	30.0.0014	33214	P1126	Sticky Memory
MCP	30.0.0015	32690	P9286	DM6700 Vestiges Removed
MCP	30.0.0016	33540	P1327	Error Messages for Unreleased
MCP	30.0.0017	33542	P1328	Foreign Printer Train Problem
MCP	30.0.0020	33540	P1327	Error Messages for Unreleased
MCP	30.0.0021	33539	D2518	Segmented String Variables

PATCH TABLE

SOFTWARE	PATCH	PRI	NOTE	DESCRIPTION
MCP	30.0.0022	33524	P1329	Special Peripheral Control
MCP	30.0.0025	33537	P1330	GETSTATUS Vs. Directory
MCP	30.0.0026	33534	P1331	READER BADMOM
MCP	30.0.0028	33536	P1333	PATHRES PBIT
MCP	30.0.0029	33667	P1334	CM Suppresses SEEK
MCP	30.0.0031	33659	D2496	Swapper Enhancements
MCP	30.0.0035	33659	D2496	Swapper Enhancements
MCP	30.0.0038	33657	P1395	DISCSTATUS Changes
MCP	30.0.0039	33656	P1396	OL PK
MCP	30.0.0040	33536	P1333	PATHRES PBIT
MCP	30.0.0041	33654	P1394	FILEKIND=FIRMWARE
MCP	30.0.0043	33659	D2496	Swapper Enhancements
MCP	30.0.0044	33638	P1383	Job Scheduletime
MCP	30.0.0045	33524	P1329	Special Peripheral Control
MCP	30.0.0046	33637	P1384	Multiprocessor Dumps
MCP	30.0.0049	33765	D2514	Parameters to Jobs Allowed
MCP	30.0.0050	33636	P1397	PB MT<number> Vs. ACMAX
MCP	30.0.0051	33632	P1385	Allow Recovery Files After DS
MCP	30.0.0055	33659	D2496	Swapper Enhancements
MCP	30.0.0056	33623	P1391	Direct Read from Schedule File
MCP	30.0.0057	33652	P1401	UA Path to Tape
MCP	30.0.0058	33651	P1408	FAULTHANDLER Vs. Tagged Words
MCP	30.0.0060	33659	D2496	Swapper Enhancements
MCP	30.0.0063	33536	P1333	PATHRES PBIT
MCP	30.0.0070	33647	P1415	Tape 403 Result Descriptor
MCP	30.0.0072	33540	P1327	Error Messages for Unreleased
MCP	30.0.0074	33866	P1457	Filedata Catalog Resident Info
MCP	30.0.0075	33646	P1432	CP Vs. AIT
MCP	30.0.0076	33645	P1433	CP Security
MCP	30.0.0077	33862	P1454	READALABEL Update of Tape Kind
MCP	30.0.0080	33856	P1478	Ending Banner Vs. Page Size
MCP	30.0.0082	33846	P1479	Protected Files Vs. Update Fil
MCP	30.0.0084	33843	P1480	Reply Left Armed (NODISK)
MCP	30.0.0085	33845	D2539	OPTIMIZER Option Removed
MCP	30.0.0086	33839	P1482	SWAPPER Vs. EI
MCP	30.0.0087	33840	P1483	Blocksearch, Epilog Vs. GOTOSO
MCP	30.0.0088	33847	P1487	FLATREADER Bad Record Message
MCP	30.0.0092	33848	P1493	LIBMAINT Vs. TAPEPARITYRETRY
MCP	30.0.0093	33834	P1512	DMSOPEN Header
MCP	30.0.0097	33635	P1233	Split DBS into Active/Nonactiv
MCP	30.0.0098	33845	D2539	OPTIMIZER Option Removed
MCP	30.0.0099	33823	P1555	Make SWAPPER Save When Running
MCP	30.0.0101	33825	P1554	RCNTL
MCP	30.0.0102	33815	P1553	Controller Malfunction
MCP	30.0.0103	33824	P1559	Volume Delete
MCP	30.0.0104	33810	P1560	AX
MCP	30.0.0105	33809	P1561	FORMATBUFFER
MCP	30.0.0106	33808	D2550	Return Queue of Origin
MCP	30.0.0108	33804	P1562	Model 4 Multiplexor
MCP	30.0.0109	33969	P1563	IAD Vs. HLUNIT
MCP	30.0.0110	33714	P1537	Dangling Reply Event in GETUSE
MCP	30.0.0112	33635	P1233	Split DBS into Active/Nonactiv
MCP	30.0.0113	33848	P1493	LIBMAINT Vs. TAPEPARITYRETRY
MCP	30.0.0115	33635	P1233	Split DBS into Active/Nonactiv
MCP	30.0.0121	34085	D2577	Close REEL Statements
MCP	30.0.0128	34096	P1224	Avoid GO TO Exited Block
MCP	30.0.0129	33085	D2579	"Pre-2.4 ON <fault> Statements
MCP	30.0.0132	33635	P1233	Split DBS into Active/Nonactiv
MCP	30.0.0133	34066	P1667	Update I/O Vs. IOERROR
MCP	30.0.0143	33539	D2518	Segmented String Variables
MCP	30.0.0148	33765	D2514	Parameters to Jobs Allowed
MCP	30.0.0163	34101	P1668	Memory Dump
MCP	30.0.0165	34261	P1659	TD830 Vs. Non-TD830 ODT
MCP	30.0.0166	33014	D2586	Test for WFL Data Decks Being
MCP	30.0.0176	33272	P1313	LINKLLISTIT
MEMORY	30.0.0001	33665	P1377	Remove BCL Characters
MEMREQ	30.0.0001	33665	P1377	Remove BCL Characters
MILO	30.0.0002	32947	P1161	Delete IO Complete Timers
MILO	30.0.0003	32948	P1162	Implement General IO Timers
MILO	30.0.0004	32950	P1163	Reduce Flow-Stopped Timers
MILO	30.0.0005	32953	P1166	Improve Disk R.D. Logging
MILO	30.0.0006	32906	P1167	ITEM-TYPE in FLOW-STOPPED Sect
MILO	30.0.0007	33413	P1249	Handshake Failure
MILO	30.0.0008	33420	P1239	Console Switches
MIL6700	30.0.0001	32768	P1189	Implement Dollar Option VERSIO
MIL6700	30.0.0002	32769	P1190	Delete Programdump Statement
MIL6700	30.0.0003	32911	P1188	INCLUDE Dollar Option
MIL6700	30.0.0004	32912	D2412	Improve Dollar Option Handling
MIL6700	30.0.0005	32768	P1189	Implement Dollar Option VERSIO
MIL6700	30.0.0006	32956	P1164	Remote Cycle and Version
MIL6700	30.0.0007	33364	D2457	Conditional OMIT
MIL6700	30.0.0009	33411	P1272	Logical Expressions

PATCH TABLE

SOFTWARE	PATCH	PRI	NOTE	DESCRIPTION
MPX	30.0.0001	33665	P1377	Remove BCL Characters
NDL	30.0.0002	31937	P9121	One-Character Request Identifi
NDL	30.0.0003	32758	P9122	EOF with Missing Define Crossh
NDL	30.0.0004	32759	P1056	Increment Sequence Corrected
NDL	30.0.0005	32762	P1361	Auxiliary Logic Correction
PATCH	30.0.0001	32816	P9123	Incorrect Handling of "$.DISK"
PATCH	30.0.0003	31585	P1198	Incorrect Sequence Error Issue
PEP	30.0.0001	33283	P1138	Load Train Tables
PEP	30.0.0002	33554	P1260	Code Space in SEG1
PEP	30.0.0003	33555	P1261	READ EXTENDED RD OPTION
PEP	30.0.0004	33556	P1262	Run with PASS Set
PEP	30.0.0005	33557	P1263	Peripheral Unit Search
PEP	30.0.0006	33665	P1377	Remove BCL Characters
PEP	30.0.0007	33711	P1670	64 Memory Modules
PEP	30.0.0008	33712	P1671	Change Printer and Display Mes
PEP	30.0.0009	33713	P1672	Specified and Nonspecified Tra
PEP	30.0.0010	34306	P1673	Multiple Processors Initialize
PEP	30.0.0011	34307	P1674	Indicate PK2 Test Case Modifie
PEP	30.0.0012	34308	P1675	No PB MAE Test if MM63 is Pres
PEP	30.0.0013	34309	P1676	Make RUNHCT Run in Control Sta
PKACTUATOR	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PKBASIC	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PKDUMP	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PKHDOVER	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PKHEADISOL	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PKINTERCH	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PKSCAN	30.0.0001	33490	P1264	Fault Printout
PKSCAN	30.0.0002	33553	P1265	Add Controlware Level
PKSEEK	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PKTEST	30.0.0001	33710	P1669	Run on BX385-235
PKWRITEREAD	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PKXD	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PK04	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PK05	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PK06	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PK07	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PK08	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PK09	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PK10	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PK11	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PK15	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PK16	30.0.0001	33707	P1442	New Controlware, Disk Pack Typ
PLI	30.0.0003	30068	D2261	LEX LEVEL > 15
PLI	30.0.0004	32685	D2265	Standardization of Compiler Fi
PLI	30.0.0005	30072	P1041	Failure to Flag Large Precisio
PLI	30.0.0007	33077	D2422	STATISTICS
PLI	30.0.0009	33073	P1302	RESTARTED Task Attribute
PLI	30.0.0015	33870	P1467	"SET STATISTICS"
PLI	30.0.0016	33871	P1468	Preprocessor INITIAL Attribute
PLI	30.0.0017	32685	D2265	Standardization of Compiler Fi
PLINTRN	30.0.0003	32521	P9125	ISAM , ISCLOSE Checks Value of
PLINTRN	30.0.0004	32519	P9126	ISAM , Delete First Record in
PLINTRN	30.0.0005	32518	P9127	ISAM , Numeric Keys
PLINTRN	30.0.0006	32520	P9128	ISAM , Adding Records to Unblo
PLINTRN	30.0.0007	32516	P9129	ISAM , Return Proper Value fro
PLINTRN	30.0.0009	32517	D2254	Summary of Conditions and DS'e
PLINTRN	30.0.0010	32871	P9130	ISAM , Erroneous ISCLOSE Resul
PLINTRN	30.0.0012	33077	D2422	STATISTICS
PLINTRN	30.0.0013	33068	P1099	Character to Bit Conversion
PRINTAUDIT	30.0.0003	32789	P9185	Elimination of Holes in Restar
PRINTAUDIT	30.0.0005	33345	P1351	Time Off by One Second
PROPERTIES	30.0.0002	32300	P8916	Expand Description Files
PROPERTIES	30.0.0005	32289	P8943	Correct Discontinuity Checking
PROPERTIES	30.0.0006	32279	D2219	Node Syntax
PROPERTIES	30.0.0009	32720	D2291	Use of Description File
PROPERTIES	30.0.0010	32280	P9264	Eliminate MASKSEARCH in FIND K
PROPERTIES	30.0.0011	32789	P9185	Elimination of Holes in Restar
PROPERTIES	30.0.0013	32781	D2288	Path Fixing Consistent
PROPERTIES	30.0.0014	32970	P1028	DASDL Update Timestamp
PROPERTIES	30.0.0015	33093	D2458	Version Overrides Recorded
PROPERTIES	30.0.0016	33250	P1182	DBS D3 NOMEM
PROPERTIES	30.0.0017	31104	D2462	Read Ahead Buffering
PROPERTIES	30.0.0018	33430	D2497	New Method of Data Base Initia
PROPERTIES	30.0.0019	33343	P1345	Retry Disk Read on Checksum Er
PROPERTIES	30.0.0020	33344	P1352	Essential, Inessential Audit R
PTNCTL	30.0.0001	32279	D2219	Node Syntax
PTNCTL	30.0.0002	32721	P1024	Change Include Range on DATABA
PTNCTL	30.0.0005	33340	P1350	Wrong Family for Global Data
READERSORTER	30.0.0001	32802	D2414	Add RSC-3 Test Name
READERSORTER	30.0.0004	32909	P1035	Modify RSP DIED Result Descrip
RECOVERY	30.0.0004	32289	P8943	Correct Discontinuity Checking
RECOVERY	30.0.0005	32279	D2219	Node Syntax

PATCH TABLE

SOFTWARE	PATCH	PRI	NOTE	DESCRIPTION
RECOVERY	30.0.0007	32290	P8975	Handle Audit Block Size Change
RECOVERY	30.0.0008	32278	P8976	Die on DBST if REBUILD
RECOVERY	30.0.0010	32276	P8977	Positioning the Audit for Reco
RECOVERY	30.0.0011	32435	P9017	NO FILE PARAMETERS
RECOVERY	30.0.0013	32791	P9181	Correct Ordered Set Paths
RECOVERY	30.0.0014	32789	P9185	Elimination of Holes in Restar
RECOVERY	30.0.0016	32985	P9254	Correct Possible INVALID INDEX
RECOVERY	30.0.0017	32781	D2288	Path Fixing Consistent
RECOVERY	30.0.0018	32720	D2291	Use of Description File
RECOVERY	30.0.0019	33093	D2458	Version Overrides Recorded
RECOVERY	30.0.0020	32961	P1209	Termination Conditions, ROLLBA
RECOVERY	30.0.0021	32789	P9185	Elimination of Holes in Restar
RECOVERY	30.0.0022	33344	P1352	Essential, Inessential Audit R
RECOVERY	30.0.0023	33341	P1353	Miscellaneous Rebuild Error
RECOVERY	30.0.0024	33335	P1354	Rebuild After Deleting Structu
RECOVERY	30.0.0025	33330	P1393	Ordered Data Set Halt/Load Rec
RECOVERY	30.0.0026	30006	P1445	Stretch Header for Current Row
RECOVERY	30.0.0028	30011	P1521	Recovery of Partitioned Struct
RECOVERY	30.0.0029	32020	P1532	UNIDENTIFIED MISCELLANEOUS ERR
RECOVERY	30.0.0030	32017	P1583	ROLLBACK Using Tape Audit
RECOVERY	30.0.0031	32001	P1614	Restart Reconstruction at Rest
RECOVERY	30.0.0032	33792	P1621	Rebuild Fails With On-Line Dum
RECOVERY	30.0.0033	33762	P1633	RECOVERY Takes Programdump on
RECOVERY	30.0.0034	33763	P1632	Erroneous Row Lockout
RECOVERY	30.0.0035	33764	P1631	ROLLBACK Gets Timestamp Mismat
RECOVERY	30.0.0036	33755	P1634	Data Recovery Setting Bad EOF
RECOVERY	30.0.0037	34284	P1696	UNIDENTIFIED MISCELLANEOUS ERR
RECOVERY	30.0.0038	34297	P1684	Halt Load Recovery Uses Wrong
RECOVERY	30.0.0039	34298	P1701	Audit Beyond BLKIMG Record Fai
RECOVERY	30.0.0040	34294	P1702	"LIMIT = THRU AUDIT" Failure
REORG	30.0.0001	32198	P8912	Sort Error #4 on Internal Sort
REORG	30.0.0002	32204	P8918	Resequence Document Section
REORG	30.0.0004	32203	P8917	Discontinuity Record on Fixed
REORG	30.0.0005	32279	D2219	Node Syntax
REORG	30.0.0007	32738	P9093	Invalid Reorganization of Empt
REORG	30.0.0008	32714	P9083	Remove Use of Certain Compile
REORG	30.0.0010	32202	P1025	Program Efficiency
REORG	30.0.0011	32727	P9183	Invalid Reorganization of Rand
REORG	30.0.0013	32731	P9255	NORECORD Error Empty Unordered
REORG	30.0.0014	32971	P1055	Correct Allocation of Restart
REORG	30.0.0015	32978	P1063	INVALID INDEX Reorganizing Bit
REORG	30.0.0018	33784	P1697	Process Time Acceleration
REORG	30.0.0019	33778	P1698	Invalid Checksum
RESHELPER	30.0.0001	33494	P1266	Add TD830 Display and DPK235
RJE	30.0.0003	32328	P9033	Usercode Attached to Card Inpu
RJE	30.0.0004	32632	D2406	Enhanced Backup File Handling
RJE	30.0.0005	32649	D2262	STATIONID Caused Reconfigurati
RJE	30.0.0006	32680	D2260	WAIT No Longer a Debug Option
RJE	30.0.0007	32621	P9216	Summary File Purge by *PB
RJE	30.0.0008	32682	D2274	SAVE/READY Station Removal
RJE	30.0.0009	32613	P1011	Usercode Attributes for RJE St
RJE	30.0.0011	32611	D2401	REMOTEPUNCH $ Option Removal
RJE	30.0.0012	32608	P1010	Blank Cards Inserted into Inpu
RJE	30.0.0013	32605	P1013	B6700 Vs. B7700 Backup Handlin
RJE	30.0.0014	32601	P1012	INVALID INDEX for >15 BD Files
RJE	30.0.0015	32587	P1045	Family Attribute Incorrect Syn
RJE	30.0.0016	32585	D2426	Device Not Ready Abort QT of P
RJE	30.0.0017	32584	P1075	Extra Page of Paper
RJE	30.0.0018	32683	D2431	Implement SF and TF on a Stati
RJE	30.0.0019	33204	P1121	REBUILDPRINTQUEUE BD File Corr
RJE	30.0.0020	33202	P1122	INVALID INDEX with One Word Me
RJE	30.0.0021	33177	P1257	Printing Jobs with More Than 1
RJE	30.0.0022	33194	D2490	"*PER" Implementation
RJE	30.0.0023	33165	P1304	USER Option for WFL Secured Re
RJE	30.0.0024	33162	D2491	LOGON Without PASSWORD on B700
RJE	30.0.0026	33175	D2492	Backup File Routing
RJE	30.0.0027	33143	P1306	RJELINKED Recovery Problem
RJE	30.0.0028	33141	D2489	LOGON Syntax Changes and Exten
RJE	30.0.0029	33139	P1303	REMOTECONTROL Vs. PRIMARYQUEUE
RJE	30.0.0030	33132	D2504	RSC TERM Defaults and Use Imp
RJE	30.0.0031	33621	P1424	SYSTEM/BACKUP Vs. Disk or Pack
RJE	30.0.0032	33718	D2529	Rjelinked File Title Change
RJE	30.0.0034	33869	P1425	Foreign Station Transfer
RJE	30.0.0035	33817	P1531	Losing Blank Card at End of BI
RJE	30.0.0036	33967	D2560	"*STATUS" Report NO TASKS ACTI
RJE	30.0.0037	33966	P1558	WFL COMPILER with No PRIORITY
RJE	30.0.0038	33965	P1566	DESTNAME NEQ RJE Terminal Pr
RJE	30.0.0039	32613	P1011	Usercode Attributes for RJE St
RJE	30.0.0040	33966	P1558	WFL COMPILER with No PRIORITY
RJE	30.0.0042	34051	P1624	BREAK on RJE RSC
RJE	30.0.0043	34050	P1625	FORMEDLP Array Vs. REMOTEPUNCH
RJE	30.0.0044	34048	P1626	Missing SKIP to Channel 1

PATCH TABLE

SOFTWARE	PATCH	PRI	NOTE	DESCRIPTION
RSLOG	30.0.0001	32953	P1166	Improve Disk R.D. Logging
RSLOG	30.0.0002	32954	P1170	Handle Multi-Record Log Entrie
RSLOG	30.0.0003	32955	P1169	Correct Date Printout
RSNETL	30.0.0001	32904	P1168	Declaration of Zero Reader/Sor
SCR	30.0.0001	33495	P1308	TESTOP Result for Display
SCR	30.0.0002	33496	P1309	TESTOP Subtype
SCR	30.0.0003	33550	P1310	Return MPXIII in UNITTYPE
SCR	30.0.0004	33551	P1311	Allow CONTBUF Read on BX385
SCR	30.0.0005	33484	P1251	SCR MARKNO III
SCR	30.0.0006	33674	P1426	Delete EXPERIMENTAL Option
SCR	30.0.0009	33708	P1495	Allow Train Table Load
SCR	30.0.0010	33845	D2539	OPTIMIZER Option Removed
SCRCONFIG	30.0.0001	33493	P1267	Add DPK235
SCRCONFIG	30.0.0002	33553	P1265	Add Controlware Level
SCTABLEGEN	30.0.0003	33521	P1427	Redundant Character Correction
SC01	30.0.0001	33468	P1227	TD830 and B7700 Update
SC02	30.0.0001	33468	P1227	TD830 and B7700 Update
SC03	30.0.0001	33468	P1227	TD830 and B7700 Update
SC04	30.0.0001	33468	P1227	TD830 and B7700 Update
SC05	30.0.0001	33468	P1227	TD830 and B7700 Update
SC06	30.0.0001	33468	P1227	TD830 and B7700 Update
SC07	30.0.0001	33468	P1227	TD830 and B7700 Update
SC08	30.0.0001	33468	P1227	TD830 and B7700 Update
SC09	30.0.0001	33468	P1227	TD830 and B7700 Update
SC10	30.0.0001	33468	P1227	TD830 and B7700 Update
SC11	30.0.0001	33468	P1227	TD830 and B7700 Update
SC12	30.0.0001	33468	P1227	TD830 and B7700 Update
SORT	30.0.0001	33634	P1398	MERGE Vs. UNITS=1
SORTERCONTRL	30.0.0001	32547	D2413	Allow PIC X in Reader/Sorter R
SORTERCONTRL	30.0.0002	32912	D2412	Improve Dollar Option Handling
SORTERCONTRL	30.0.0003	32956	P1164	Remote Cycle and Version
SORTERCONTRL	30.0.0004	32906	P1167	ITEM-TYPE in FLOW-STOPPED Sect
SORTERCONTRL	30.0.0005	33364	D2457	Conditional OMIT
SORTERCONTRL	30.0.0007	33424	D2473	Implement MOVE LENGTH
SORTERCONTRL	30.0.0008	33425	D2481	Black Band Warning
SORTERCONTRL	30.0.0009	33426	D2474	Dollar Warning Default
SORTERCONTRL	30.0.0010	33419	P1250	Picture Following SIZE
SORTERCONTRL	30.0.0011	33418	D2475	Volume/Subset Criteria
SOURCENDL	30.0.0001	32766	P9303	Modifications to TD800 Request
SUPERIV	30.0.0001	33553	P1265	Add Controlware Level
SUPERIV	30.0.0002	33670	P1436	Error Retry for New Controlwar
SUPERIV	30.0.0003	33671	P1437	Balance of Track Read Length
TABLEGEN	30.0.0002	30070	D2427	String Implementation
TABLEGEN	30.0.0003	30070	D2427	String Implementation
TABLEGEN	30.0.0004	32545	D2400	TABLEGEN No Longer Handles ALG
TAPEHANDLER	30.0.0001	33399	P1270	List Card Assembly Part Number
TAPEHANDLER	30.0.0002	33511	P1448	New Data File Title = Old File
TAPEHANDLER	30.0.0004	33511	P1448	New Data File Title = Old File
TAPEHANDLER	30.0.0007	33579	P1440	Deleted FAMILY Statements
TAPEHANDLER	30.0.0010	33511	P1448	New Data File Title = Old File
TAPEHANDLER	30.0.0011	34173	P1677	Maintenance Kit Release Versio
TAPEHANDLER	30.0.0012	33408	P1704	Update Comments in Symbolic
TIMERTEST	30.0.0001	33665	P1377	Remove BCL Characters
TPE	30.0.0001	33227	P1116	Title Change
TPE	30.0.0002	33665	P1377	Remove BCL Characters
TP64	30.0.0001	33709	P1496	SCR Tests for Train Printers
TP64A	30.0.0001	33709	P1496	SCR Tests for Train Printers
TP72	30.0.0001	33709	P1496	SCR Tests for Train Printers
TP96F	30.0.0001	33709	P1496	SCR Tests for Train Printers
TP96S	30.0.0001	33709	P1496	SCR Tests for Train Printers
UNCOLINT	30.0.0001	32547	D2413	Allow PIC X in Reader/Sorter R
UNCOLINT	30.0.0003	33424	D2473	Implement MOVE LENGTH
UTIL	30.0.0002	32201	P8944	New Tape Block Default Size
UTIL	30.0.0004	32279	D2219	Node Syntax
UTIL	30.0.0007	32286	P8985	Reconstruct Onto DB Files
UTIL	30.0.0008	32443	P9019	Reconstruct Empty File
UTIL	30.0.0010	32440	P9076	No Reconstruction on Row Not D
UTIL	30.0.0011	32418	D2244	Limit Error
UTIL	30.0.0012	32740	D2243	Specify PACKNAME
UTIL	30.0.0013	32710	P9182	No File for Dumps to Disk
UTIL	30.0.0014	32711	P9184	New III.0 Format Tape Blocks
UTIL	30.0.0016	32737	P9258	READERROR Set for Selected Row
UTIL	30.0.0017	32736	P9279	Problems in RECONINFO
UTIL	30.0.0018	32735	P9278	Problems with Partitions
UTIL	30.0.0019	32972	P1072	File Listed Inside Parentheses
UTIL	30.0.0020	32977	D2436	Halt/Load Bit Override
UTIL	30.0.0021	33093	D2458	Version Overrides Recorded
UTIL	30.0.0022	33100	P1210	Correct Blocksize for CFOLD
UTIL	30.0.0023	32973	P1212	Correct Allocation of Tapes
UTIL	30.0.0024	32974	P1211	Label BADTAPE
UTIL	30.0.0025	33349	P1278	Recovery using II.8 Dump Tapes
UTIL	30.0.0026	33348	D2482	No File on CONTROLOLD

PATCH TABLE

SOFTWARE	PATCH	PRI	NOTE	DESCRIPTION
UTIL	30.0.0027	33430	D2497	New Method of Data Base Initia
UTIL	30.0.0028	33347	P1355	Remove First Read Error Messag
UTIL	30.0.0029	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0030	33108	P1356	Fault Initializing Large Name
UTIL	30.0.0031	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0032	33336	P1349	Check for Valid Structure Numb
UTIL	30.0.0033	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0034	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0035	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0036	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0037	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0038	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0039	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0041	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0042	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0043	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0044	30009	P1581	Halt/Load Restart with Multipl
UTIL	30.0.0045	32004	P1597	Tape Kind Set Correctly for Fi
UTIL	30.0.0046	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0047	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0048	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0049	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0050	33790	P1622	Row Selection with Multiple Op
UTIL	30.0.0051	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0052	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0053	33757	P1630	Fault in REALSWITCH
UTIL	30.0.0054	33799	P1629	System Partitioned Files can b
UTIL	30.0.0055	33789	P1635	PRINTIT Gives Error for Block
UTIL	30.0.0057	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0058	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0059	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0060	34291	P1699	Initializing Standard Data Set
UTIL	30.0.0062	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0063	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0064	33338	D2441	Multiple Worker Recovery
UTIL	30.0.0065	33338	D2441	Multiple Worker Recovery
WFL	30.0.0006	32873	P9131	WFL Termination if DS While Sc
WFL	30.0.0008	32879	P9132	Error for Missing UNTIL
WFL	30.0.0009	32877	P9133	Eliminate INVALID INDEX on WFL
WFL	30.0.0010	32876	P9134	Proper Information for Long CO
WFL	30.0.0011	32875	P9135	Perform Job Rollout in More In
WFL	30.0.0012	32880	P9159	Information Carry Over
WFL	30.0.0013	32888	D2266	Limit on Nested Subroutine Dec
WFL	30.0.0014	32890	P9247	DECK STATEMENT
WFL	30.0.0017	32887	D2423	AREASIZE and BLOCKSIZE Attribu
WFL	30.0.0018	32613	P1011	Usercode Attributes for RJE St
WFL	30.0.0019	32891	P1046	Explicit Loops around COMPILE
WFL	30.0.0020	32894	P1047	Generate Error for Bad KIND Sp
WFL	30.0.0021	32892	P1048	WFL Aborts for Extremely Large
WFL	30.0.0022	32893	P1049	String Expressions in ON State
WFL	30.0.0023	33036	P1050	Handling of KIND Lists in OLD
WFL	30.0.0024	33038	P1051	DATA or END Within File Names
WFL	30.0.0025	33024	P1052	Handling of Missing ?END JOB C
WFL	30.0.0026	33025	P1064	Improve Handling End of File
WFL	30.0.0027	33037	P1068	Infinite Loop Following Securi
WFL	30.0.0028	32889	D2434	Clearing Card Reader in Use by
WFL	30.0.0030	33030	P1067	Eliminate WFL Abort on Bad Fil
WFL	30.0.0031	33029	D2433	LINEINFO for Job Code Files
WFL	30.0.0032	33026	P1066	Long COPY Statements
WFL	30.0.0033	33027	P1065	Single START Statement
WFL	30.0.0035	33032	D2449	Expressions in COMPILE Stateme
WFL	30.0.0037	33765	D2514	Parameters to Jobs Allowed
WFL	30.0.0038	33765	D2514	Parameters to Jobs Allowed
WFL	30.0.0039	33766	P1376	Errors in Declarations
WFL	30.0.0042	32887	D2423	AREASIZE and BLOCKSIZE Attribu
WFL	30.0.0043	32892	P1048	WFL Aborts for Extremely Large
WFL	30.0.0045	33693	P1485	Boolean Constant Evaluation
WFL	30.0.0046	33692	P1489	Correct HEX Function
WFL	30.0.0047	33767	P1488	Single Blank Character String
WFL	30.0.0048	33765	D2514	Parameters to Jobs Allowed
WFL	30.0.0049	32887	D2423	AREASIZE and BLOCKSIZE Attribu
WFL	30.0.0052	33765	D2514	Parameters to Jobs Allowed
WFL	30.0.0053	33896	P1658	Backward Branches in ON Statem
WFL	30.0.0054	33015	P1660	$INCLUDE with NEWSOURCE or LIB
XREFANALY	30.0.0001	30070	D2427	String Implementation
XREFANALY	30.0.0002	30070	D2427	String Implementation
XREFANALY	30.0.0003	32590	D2425	XREF Procedure End Sequence Nu
XREFANALY	30.0.0005	33701	P1429	Correction Declaration Count
XREFANALY	30.0.0006	33700	D2526	Change INTERACTIVEXREF File Ti