UT06/5 Program Generator

A. Description

UT06 will process data in accordance with instructions supplied on cards. The data may be read from cards, tape or disk files, and written to tape or disk files or printer.

B. Operating Instructions

---omitted---

D. Instruction Layout Notes

1. Instructions may be punched into any columns of a card.

2. A word is terminated by one or more spaces and/or commas, unless enclosed in quotes.

3. A word or literal may not be split over two cards.

4. A paragraph name must start with a numeric character, and the first 16 characters must be unique. A paragraph name may start in any column of a card.

E. Notation Used in Instruction Descriptions

1. If clauses are enclosed in braces { } one clause must be chosen.

2. Clauses enclosed in square brackets [ ] are optional.

3. If a word in capital letters is used, only the underlined characters are mandatory. In almost all cases, two characters only are required.
(Note: underlining has not been done in this Notepad document)

4. Three full stops ... indicates that the preceding clause may be repeated indefinitely.

F. Definition Of Terms

Character:
- any of the 64 character ICL set.

Word:
- a string of characters terminated by space, comma, fullstop/space, or column 80 of card.
- maximum length 80 characters

Alphanumeric Literal:
- a string of characters enclosed in double quotes "
- The string may include any character except double quote"
- If column 80 of the card is reached before the second quote is found then col. 80 will terminate the literal.
- maximum length excluding quotes 79 characters.
- terminated as WORD above.

Numeric Literal:
- a string of numeric characters (ie: in the set 0 to 9)
- may be preceded by a minus sign.
- maximum length 13 digits.
- terminated as WORD above.

Statement:
- consists of a KEYWORD followed by words or literals as defined for the keyword used.
- terminated by full-stop, another keyword, or **** card.

Sentence:
- one or more statements, the last of which is terminated by a full-stop.
- a sentence containing the keyword IF must be terminated by a full-stop before another IF or end of source program.

Source Program:
- cards containing statements
- terminated by at least two consecutive asterisks ** anywhere on a card.

Card Data:
- follows source program which was terminated by only two asterisks **
- terminated by ****
- no more cards may be read after **** card.

G. Validation

All source words are validated and are flagged with an arrow if in error. If an error is detected, the program will halt after checking all cards, and the object program cannot be run.

H. Data Definition

Data fields are defined as they are used by specifying a record area denoted by the characters R, W, C, P, H, I, J, A, and by the word and/or character position within the area -

EG: word 1 of Record area = R1
word 2 character 3 of Work area = W2.3

If the length of the field is not 1 word (4 characters) and is not associated with an alphanumeric literal, the length should be specified by /n where n is the number of characters, eg: R1/12.

Data may be held in any of the following areas (each area being denoted by an alphabetic letter):

R Record area. 200 words, initially zero.
Tape and disc records are read into and written from area R.
First word = R, last word = R199.

W Work area. 100 words, initially zero.
Used for temporary storage of records, totals etc.
First word = W, last word = W99.

C Card area. 80 characters, initially spaces.
Cards are read into this area.
First character (card column 1) = Cl, last = C80.

P Print area. 160 characters, initially spaces.
Print position 1 = P1, print position 160 = P160.
To print data moved to this area use PRINT instruction.
The area is space filled after printing.
Three headings are printed automatically (see areas H, I, J).
It is possible to alter details such as number of headings, lines per page, channel numbers of headings or print line, page count etc by using area A.

H First heading. 160 characters, initially containing standard program identification, date and page number starting at 1, printed at CHANNEL-1.
Print position 1 = H1, pp160 = H160.

I Second heading. 161 characters. First character I = spacing (preset to "2", may be set to "1"), remainder of area initially spaces.
Print position 1 = I1, pp160 = I160.

J Third heading. 161 characters. First character J = spacing (preset to "1", may be set to "2"), remainder of area initially spaces.
Print position 1 = J1, pp160 = J160.

A A510 area. 170 words (includes areas P, H, I, J).
A = lines per page., initially 56.

Al = lines printed since start of run.
A2 = lines printed this page initially 100.
A3 = page no.
A4 = no. of headings, initially 3, maximum 3.
A6 = printline spacing, initially "SP 2".
A47 = first heading spacing, initially "HEAD"
A88 = second heading spacing, initially "SP 2"
A129 = third heading spacing, initially "SP l"

I. File Definition (IN,OUT)

Files to be read or written to are defined by means of IN or OUT statements respectively. These statements may appear anywhere in the set of instructions. The clauses following IN/OUT may be in any order.

Multiple input files

If more than one IN statement is present, the files defined by IN statements will, be read one at a time in order of definition. At the end of the last input file the AT END action will be taken. (See READ verb).
A file is opened automatically when the first READ or WRITE statement for that file is actioned, and is closed at end of run or when another file of the same mode (IN or OUT) has a READ or WRITE actioned.
It is not possible to create an empty file unless the source program consists of a single OUT statement (no labels or verbs allowed).
Files may be differentiated by using file numbers in the range 0 - 15. If not specified, 0 is assumed. When reading, all input files with the file number specified in the read statement (or zero) will be read once, in the order of the IN statements, before the AT END statement takes effect.

Multiple Reading of Files

A file may be read an unlimited number of times by issuing another read after end of file. Note that if the AT END clause is not specified, the run will terminate at end of file.

Output files

Only one OUT statement is permitted for each file number.

Default values

Any or all of the clauses following IN/OUT may be omitted.

The default values specified below then apply.

Maximum number of files

The total number of IN and OUT statements allowable is 15.

Format:

Opening Mode	Clause	Clause Description	Default
IN	[MTn]	File medium	MT
OUT	[EDn]	n=file number	
	[DAn]	(may be omitted if zero)	
	[alphanuineric-literal]	File name	"UT06-OUTPUT"
	[GEN numeric-literal]	Generation No.	GEN 0
	[BLOCK numeric-literal]	Block size (characters)	BLOCK 2048

Notes

1. If generation no. for input tape is 0 or the clause is omitted, a tape with any generation no. will be read.

2. Block size is ignored unless for output MT, when the size stated will be rounded up to one of the values 64, 128, 512 or 2048 characters. If not specified, 2048 is assumed.

3. If SAME follows IN or OUT, details of the previous file will be copied, and may be altered with further clauses.

Examples

Statement	Medium	Filename	Gen	Block
IN "DEBTOR-MASTER"	MT	DEBTOR-MASTER	0	2048
OUT "DEBTOR-INDEX"	MT	DEBTOR-INDEX	0	2048
IN ED "P2D-VALIDIN"	ED	P2D-VALIDIN	0	2048
OUT "TEST DATA" BLOCK 128 GEN 1	MT	TEST DATA	1	128
OUT	MT	UT06-OUTPUT	0	2048
OU BL 1024 GE 0 "P2X-WORK" ED	ED	P2X-WORK	0	2048
OUT MT "TEST", GENERATION-NO 3.	MT	TEST	3	2048

J. INPUT/OUTPUT VERBS (READ, WRITE, PRINT)

General notes

All files use the same record area R except cards or printer. If a record of more than 200 words is read, the program will abandon.

READ

Reads input files in the order specified, one at a time. At end of final input file, AT END action will be taken if present, otherwise all files are closed and program finishes. If not end of file, and AT END is specified, control passes to the beginning of the next sentence, or ELSE statement. To read a card, replace Fn below by CARD.
Cards must follow instructions.
Terminate instructions with **
Terminate cards with ****

Format 1: READ [Fn].

Format 2: READ [Fn] [AT] END statement-1 [statement-2... statement-n]. Where n is the file number specified in the IN statement (Fn may be omitted if n = zero).

Examples

```
1RD. READ, AT END MOVE 99 TO R1, WRITE, STOP. WRITE G0 TO 1RD.
READ. IF R1 = "AOOO" MOVE R2/20 TO W, READ.
```

WRITE

Writes a record to the output file.
The record (and record word count) remains the same after the write.
If the record word count is zero, it will be set to 21 before writing
(for when copying 80 character cards).

Format: WRITE [Fn]

Where n is the file number specified in the OUT statement. (Fn may be omitted if n = zero).

PRINT

Prints 160 characters of area P(P1-P16O) with single line spacing.
Up to 3 headings are printed.

Heading areas are H, I, and J. These areas must be set up before printing if required.

```
eg:     MOVE "LISTING OF FIlE XXX" TO H10
        MOVE "RWC BRANCH CUST" TO I1
```

To print single line spacing MOVE "l" to P
To print double line spacing MOVE "2" to P

```
Format: PRINT
```

K. DATA TRANSFER (MOVE)

Character move

A string of between 1 and 511 characters may be moved. The number of characters to be moved is the length of the first field. The length of the second field is ignored. There is therefore no automatic spacefill or truncation as in COBOL.

```
Format: MOVE    {char-field-1} [TO] char-field-2
                {char-literal}
```

Examples

1. Move 20 characters from record word 5 to word 4.1
   MOVE R5/20 TO R4.1.

2. Move 4 characters from record word 3 to word 0.
   MOVE R3 to R (or MOVE R3.0/4 TO R0.0/4)

Decimal/Binary conversion

A string of 1 to 13 characters nay be converted to a binary field length 1 to 8 characters and vice versa.
Negative values will be converted only if the binary field is
4 or 8 characters long.

```
Format 1: MOVE  {char-field-1} [TO] binary-field-2
                {char-literal}

Format 2: MOVE binary-field-3 [TO] char-field-4
```

Example

Convert record word 3.1, 6 characters, to binary, in word 5.

```
MOVE R3.1/6 TO R5B
```

Binary to Binary Move

```
Format: MOVE    {binary-field-1} [TO] binary-field-2
                {numeric-literal}
```

Examples

```
MOVE 10 TO R1
MOVE 10 TO R1B
MOVE R1B TO R5/8B
MOVE R1.2/lB TO R5B
```

L. CONDITIONAL STATEMENT (IF)

If the condition specified is true, control passes to the statement following the condition, otherwise control passes to the start of the next sentence or ELSE statement. A sentence is terminated by a full stop followed by a space.

There may be up to 199 statements in a sentence. Field sizes between 1 and 511 characters may be compared. The length to be compared is the length of the first field, unless the second field is a literal.

If one field is a numeric literal, the other is assumed to be a one word binary field (the length is ignored if specified).

There may not be more than one IF in a sentence.

Up to 199 conditions may be tested in one sentence, using AND or OR, but AND or OR must not be mixed in the same sentence.

Format:

```
IF {char-field-1} relation {char-field-2} statement-1 [st-2...].
   {char-literal}          {char-literal-2}
```

where relation is one of the following:-

<	NOT <	LESS THAN	NOT LESS THAN	L	NOT L
=	NOT =	EQUAL TO	NOT EQUAL TO	E	NOT E
>	NOT >	GREATER THAN	NOT GREATER THAN	G	NOT G

Examples

1. If record word 2.3 = "A" move "B" in its place:

```
        IF R2.3 = "A" MOVE "B" TO R2.3.
```

2. If record word 4 is less than word 5 write a record and stop.

```
        IF R4 < R5 WRITE STOP.
```

3. IF R1 = R2 AND R3 < R4 AND > R6 GO TO 1 READ.

4. IF R5.2 = "AA" OR "B" OR "KA" OR R5.3 = "Z" GO TO 1-EXIT.

M. CONTROL TRANSFER (GO TO, PERFORM, STOP, CALL)

General note

Paragraph names must start with a numeric character (but may be anywhere on a line)

GO TO

Control is passed to the named paragraph.

Format: GO [TO] paragraph-name.

PERFORM

Control is passed to the named paragraph. At the end of the paragraph, control returns to the verb following the PERFORM statement. The COBOL restrictions on branching out of a performed paragraph apply.

Note there is no PERFORM THRU option. Any level of nesting is allowed.

Format: PERFORM paragraph-name.

STOP

All files are closed and the program halts 'END OF RUN'.

```
Format:         STOP
```

Examples.

```
READ, AT END PERFORM 2-LAST, STOP. GO TO 5-WRITE.

2-LAST. MOVE "99" TO R1, WRITE. 5-WRITE. WRITE.
```

CALL

A subroutine already linked into UTO6 may be CALLed. No parameters are allowed.

Format: CALL subroutine-name.

N. ARITHMETIC VERBS. (ADD, SUBTRACT, MULTIPLY, DIVIDE)

General notes

Data used for arithmetic must be in binary form, length from 1 to 8 characters (UT06/5 only). (Note: UT06/4A allows only length 4 chars)

Data not in this form must first be converted using the MOVE verb.

Fields of length 4 or 8 characters may be signed.

Format:

```
{ADD}                                   [TO]
{SUBTRACT}      {binary-field-1}        [FROM]  binary-field-2
{MULTIPLY}      {binary-literal}        [BY]
{DIVIDE}                                [INTO]
```

Notes

The verb is performed on field-1 and field-2 and the result is placed into field-2.

Examples

1. Add word 5 of the record to word 4.

```
        ADD R5 TO R4.
```

2. Add value held in 6 characters from word 5.1 of record, to word 3 of record, using word 1 of work area.

```
        MOVE R5.1/6 TO W1B ADD W1 TO R3.
```

MODIFICATION OF DATA LOCATIONS

The data locations accessed by instructions may be modified at run time by means of the AUGMENT instruction. (temporarily withdrawn)