30E20

COMPILED RUNNING DICTATOR

By: K. Davis and T. Baumeister III

C R U D

JAZ TO DICTATOR TRANSLATOR

The CRUD translator was developed to provide a unique, time-saving procedure for programming technical problems of small to moderate size. It permits altering the usual programming sequence to one wherein the problem, expressed in essentially algebraic notation, can be solved and the method of solution can be screened for errors and evaluated for utility before the coding, i.e. translation, step is taken. The JAZ interpreter itself provides the programmer with the simplest and most understandable program language yet devised for the LGP-30. In addition, JAZ detects both coding and logical errors at the time they are made, and by a type-out, directs the programmer to the proper remedial action. Simultaneous with the production of a problem solution, a working program tape can be prepared in the original, easily understood JAZ language. If additional problems are to be solved with this program, however, higher computational speed is desirable.

CRUD accomplishes this by translating a JAZ language program directly into a running program for one of the Royal McBee floating point interpretive systems, DICTATOR. The resulting program runs about twenty times as fast in DICTATOR as the original program in JAZ. Translation time is dependent largely upon the number of operations, the nature of the operations, and the number of data and constant storages. An average rate is 10 min/100 JAZ words. The maximum time is fixed by the available storage of 1400 words for the translated DICTATOR program.

The attached instructions, tables, and notes cover:

1. Instructions for modifying the JAZ program.
2. CRUD operating instructions.
3. Instructions for running the DICTATOR program.
4. A table of JAZ operations versus DICTATOR instructions.
5. Notes on conventions and restrictions.

CHANGING JAZ TAPES FOR TRANSLATION TO DICTATOR

No changes to a JAZ tape are required for translation to DICTATOR

if:

(1) you elect to type in manually the word ssssss' signaling "End of Program",
(2) and the last operation on the JAZ tape is s' o' or go' to some existing connector,
(3) and the JAZ tape contains all the data and constants needed to run the problem(s) with the DICTATOR tape. This is most unlikely if more than one problem or multiple cases of the same problem are to be run with the DICTATOR tape.

To meet the above conditions, changes to the JAZ tape are made in the following way:

(1) End of program signal -

Either: manually type in ssssss' as explained in the operating instructions

or: punch the word ssssss' as the last item on the JAZ tape.

(2) Last operation on JAZ tape -

Punch the pseudo-op, o' , if the last operation is not already o' , s' , or go' to some existing connector.

The o' translates as a jump (0001'00000600') to the first interpretive instruction in your DICTATOR program. It (or the s' or go') is needed as a termination to protect you when the DICTATOR program is run. Otherwise, the program as stored in memory, may continue past the last operation, and into spurious instructions stored there by some previous program. In the absence of this protection, the DICTATOR program may run improperly.

(3) Data input -

Provision for supplying the DICTATOR program with data needed anew in each run of the program is made in the following way -

(3) Data input (continued)

a) Delete from the JAZ tape the operations defining the variable data.

```
;'  '1'  'de'a'
;'  '2'.'2'  'de'bb'de'ccc'     (Example #1)
etc.
```

cannot be used to input variable data, only for constants.

The data are usually typed in or are on a separate tape which (in the JAZ run) is put through the reader before the JAZ program tape.

b) Punch a new data tape with an i' operation for each item of input data in the format -

```
i'abcde'                  (Example #2)
```

Example #1, above, is written:

```
i'a'
i'bb'
;'bb'de'ccc' etc.
```

c) Place the input data tape with i' operations ahead of the JAZ program when doing Step 2 (Translation) of the operating instructions.

d) Supply the number itself by type-in while running the DICTATOR program - see <u>Running JAZ Translated DICTATOR Program</u> - <u>Step 2</u>.

(3) Data input (continued)

The i' order is interpreted by the JAZ translator as follows -

1. Make storage (say 0156) for the thing named abcde in example #2, above.

2. Create DICTATOR printing instructions that will type out the name so we can tell what thing is to be input,

```
     e.g.          80090057
                   80090005
                   80090053
                   80090021
                   80090037

                     etc.
```

3. Create the DICTATOR instructions to input and store the number at running time,

```
     e.g.          80070156
```

The DICTATOR coding for example #1 might look like this -

```
                   80090057
                   80070306
                   80090005
                   80090005
                   80070308
                   80090053
                   80090053
                   80090053
                   80070310
```

CRUD OPERATING INSTRUCTIONS

PURPOSE

To permit a program to be written, run, and debugged easily in JAZ language, and then translated automatically into a fast running DICTATOR program for repetitive use.

INPUT

A JAZ program tape debugged and modified, if desired, with "i" orders which permit input of variable data.

OUTPUT

A complete program tape ready to run in DICTATOR.

METHOD

1. Read CRUD into memory.
2. Put your JAZ program tape in the photo reader.
3. Turn the photo reader switch to "reader".
4. Turn on the punch.
5. Push the "six bit" button.
6. Do an N .
7. At this time your DICTATOR program will be punched (see also Error Traps 1, 2, and 4).
8. When your JAZ tape runs out of the reader, turn off the punch, turn the photo reader switch to typewriter, push "start read", type in "ssssss", turn on the punch, and hit "start compute".
9. Your table of constants and connectors will be punched (see Error Trap 3).
10. Tape feed your new tape through the punch, tear it off, label it, and keep it.
11. Pick your JAZ tape out of the tape bin and keep it.
12. Get off the machine.

CRUD ERROR TRAPS

1. PROGRAM TOO BIG

   The routine is too long. It started at 600 and has reached 2000 - the end of the space available. The program must be altered to fit. If "start compute" is hit, compile (Translation) will continue typing out this statement - every line. The resulting program will not run.

2. THERE ARE 2 OF abc RECOMPILE

   You have a connector named abc in two places. The program must be altered to remove the duplication.

   If "start compute" is hit, compile will continue. The resulting program will use the first abc, but can possibly be repaired by hand to use the second if the error is one of transcription rather than logic.

3. NO CONNECTOR abc RECOMPILE

   You have had go'abc' , mem'abc' , tape'abc' , or if'abc' , but in the course of compile, you have no connector named abc to go to. This program can possibly be fixed by a competent programmer; otherwise, the JAZ program must be altered and retranslated.

   If "start compute" is hit, compile will continue. The resulting program probably will not run.

4. There are several other traps such as "GOOF" and "OVERFLOW" meaning that included constants are misspelled or too big. If start is hit, next word is taken as an operation. These can possibly be treated as in JAZ, but for the best results, your JAZ program tape should be fixed.

5. The illegitimate arithmetic operations such as log of zero do not show up at compile time. They will show up at running time and must be handled under the rules of DICTATOR.

6. TABLE TOO BIG

   You have used more named values in the table than JAZ or CRUD in its present form can handle. Rewrite your JAZ program to use fewer named values.

7. TOO MANY CONNECTORS

   You have used more than 45 connectors, and have overflowed the space allocated by CRUD.

CRUD Error Traps (continued)

8. TOO MANY CONSTANTS

You have used more then 50 constants. CRUD with its present allocation cannot handle this.

9. BUM op abc

An operation, e.g. abc, not recognizable by JAZ has been found. This can be caused by misspelled constants or missing "go"s ahead of connectors.

RUNNING JAZ TRANSLATED DICTATOR PROGRAM

After a JAZ program has been translated into DICTATOR, it has been completely altered and must there-after be handled under the rules of DICTATOR. Only a few of these rules are of interest to us:

1. To read in your DICTATOR program, the DICTATOR interpreter must be in the machine. Put your tape in the reader. Do an N . Your tape and tables of con-nectors and constants will be read in. When it is finished, go on "manual input" and do another N . Type in 30000600. Hit "start compute". Either item 2 or 3 below will occur.

2. If you had any "i" orders in your modi-fied JAZ program, the name of the piece of data required will be typed out and the "input" light will go on. Type in the one number in DICTATOR form. (See below for details.) Then type "exit". Hit "start compute". Repeat this as often as names are typed out and the "input" light goes on.

3. After the last type-in, or if you have had no "i" orders, the program will automatically proceed with the calculations.

DICTATOR

FLOATING POINT INTERPRETIVE SYSTEM

DATA INPUT FORMAT

The data input routine permits the entry of a decimal number consisting of up to eight digits. To specify this number, the following items are required:

1. The eight decimal digits of the number.
2. The sign of the eight digit number.
3. A decimal exponent to indicate the location of the decimal point associated with the eight digit number.
4. An end of data item signal.

The following data input format illustrates these four items and is the basic data input format.

$X_1\ X_2\ X_3\ X_4\ X_5\ X_6\ X_7\ X_8'\quad \pm\ E_1\ E_2$ exit'   where:

(1) $X_1\ X_2\ X_3\ X_4\ X_5\ X_6\ X_7\ X_8'$ represents an eight digit decimal with the decimal point assumed between X_1 and X_2.

(2) $\pm$ (immediately following X_8') represents the sign of the eight digit decimal.

(3) $E_1\ E_2$ represents 50 added to the power of 10 by which the decimal $(X_1\ X_2\ X_3\ X_4\ X_5\ X_6\ X_7\ X_8)$ must be multiplied to produce the number being entered.

Range: $00 \leq EE \leq 99$

e.g. 14 = 14000000' + 51'
-14 = 14000000' - 51'
1.4 = 14000000' + 50'
.0014 = 14000000' + 47'

(4) exit' signals end of data item.

LIST OF CORRESPONDENCES

JAZ TO DICTATOR

NEW (nonJAZ) COMMANDS

JAZ	DICTATOR EQUIVALENT	NOTE
Clear Counter	CRUD 10000600'	(7)
i'abc'	800900XX' 800900YY' 800900ZZ' 8007XXXX'	(3) (2) (1)
ssssss'	(End of program. The contents of the tables are punched out in hex and the translation is finished.)	(5) (6)

JAZ COMMANDS HAVING NO MEANING TO THE COMPILER

-00000	(nothing)	(8)
rfl	(nothing)	(8)
hit "start compute" (no type-in)	(nothing)	(8)
pt	(nothing)	(8)
pthx	(nothing)	(8)
rs	(nothing)	(8)

PSEUDO OPS

de	Gives ------ZZZZ	(2)
pd	18'00000000'	
t	80090016'	
s	80080004'	
o	1'00000600'	(5)
pa	800900XX' 800900YY 800900ZZ 800900AA 800900BB	(3)

JAZ RIGHT OPS

JAZ	DICTATOR EQUIVALENT	NOTE
;	Gives XXXX	(2)
+	1XXXX'YYYYZZZZ'	(2)
-	2XXXX'YYYYZZZZ'	(2)
*	3XXXX'YYYYZZZZ'	(2)
/	4XXXX'YYYYZZZZ'	(2)
q	4YYYY'XXXXZZZZ'	(2)
p	-(12'XXXX0000') (30000'YYYY0000')- (11'0000ZZZZ)	(2)
r	-(12'XXXX0000') (40000'YYYY0000')- (11'0000ZZZZ')	(2)

JAZ LEFT OPS

cs	3XXXX'0000ZZZZ'	
si	14'XXXXZZZZ'	
co	15'XXXXZZZZ'	
at	16'XXXXZZZZ'	
ln	12'XXXXZZZZ'	
aln	11'XXXXZZZZ'	
log-	(12'XXXX0000' (30000'0004ZZZZ'	
alog-	(3XXXX'00060000' (11'0000ZZZZ'	
sr	13'XXXXZZZZ'	
sq	3XXXX'XXXXZZZZ'	

GO OPS

if	2'XXXXYYYY'	(2) (4)
go	1'0000XXXX'	(2) (4)
mem	1'0000XXXX'	(2) (4)
tape	1'0000XXXX'	(2) (4)

CRUD NOTES

1. An i' order is for input of data which are to be supplied at running time. These are usually handled in the JAZ program by a separate JAZ tape with

   ;' '1'.'2' 'de'abc'

   and the like on it. Such a tape would be repunched

   i'abc'

   It is not necessary to supply constants, as opposed to data, in this manner. They will be stored in the table by CRUD.

2. All commands which refer to a constant or a name in the table, such as right ops, some pseudo ops, and de and i commands will have that constant stored or space assigned in the table. The address of that space in the table will be punched in conjunction with the appropriate DICTATOR command.

   For example:

   i'apd'

   If apd goes into the table at 0304, then the output will be

   80070304'

   This would be shown in the list of correspondence as

   8007XXXX'

3. pa and i orders provide alphabetic print-outs.

   The print control orders for DICTATOR corresponding to what is to be printed will be punched.

   The prints precede the i input order so that you can tell what piece of data is desired by the input order which follows. The input must, of course, be in DICTATOR format.

4. A connector is defined in JAZ as being the next word on the tape after a go type operation which matches the "where" of

   go'where'

For CRUD, a connector is defined more rigidly. It is defined as being the word following a "where".

For example:

```
        go'where'next'
      tape'where'next'
       mem'where'next'
```

"next" is by definition a connector.

If the word is -00000, the word after is the connector.

e.g.: mem'where'-00000'next'

5. An o command, which in JAZ corresponds to an N , and is almost never used except to go back to the beginning of a tape, is translated into a l'00000600'. At 0600 is the first command in your DICTATOR program, and corresponds to your first JAZ order.

   If your JAZ program does not end with a "go" operation, it is a good idea to type in an o before you type in the ssssss to return to the start of the routine.

6. ssssss is the signal to the Translator that no more JAZ coding is coming. At this time, the Translator searches your table for undefined connectors. (If any are found, error printout "No Connector abc" is typed - see Error Trap #3). Tables are punched out and translation is finished.

7. These are the words that are punched out by the Translator at the start of a compile before the JAZ tape is read in. They serve to tell the DICTATOR that the program will start at 0600. So the first DICTATOR command corresponding to the first thing on your JAZ tape will be in location 0600.

8. These are commands that have no significance to the Translator.