HARVEST CONTROL PROGRAM

Parameter Specification

Richard T. Burch
J. Paul Menehan
MPRO-1Ø4

Ø5NOV6Ø

I. General Information

A. Parameter Card Formats

Program parameters will be specified on Parameter Cards which accompany the request for each problem program. These cards have the following format:

Cols. 1- 5:	MPRO Job Number
Cols. 6- 7:	Step Number Within Job
Col. 8 :	J ("Job" Card)
Cols. 9-10:	PP (Problem Program Card)
Col. 11 :	B (Parameter Card)
Cols. 12-80:	Parameter Specification

B. Parameter Specification and Handling

Parameters are specified by a parameter identification and the absolute parameter or symbolic name for the parameter (e.g., ALPHSIZE = 26 or CONVTABL(S) = ENGLISH; where ALPHSIZE and CONVTABL are parameter identifications, 26 is an absolute parameter, and ENGLISH is a symbolic parameter). These parameters must be translated into absolute machine language strings of bits and absolute starting bit addresses of the memory areas in the program into which the parameters are to be inserted. This translation and insertion is done by the Problem Program File Generator by means of a Symbol Translator Table which is part of the problem program.

The Symbol Translator Table for a particular program is constructed by the Harvest Assembly Program (HAP) from Parameter Location Entry (PLE) and Symbolic Parameter Entry (SPE) pseudo-op statements. Each statement causes a two-word entry to be made in the table. The first word of each entry contains the symbol in Harvest Character Set (HCS) coding; this word is used to equate the symbol appearing on the Parameter Card with its corresponding Symbol Translator Table entry. The second word of the Parameter Location Entries (PLE) in the Symbol Translator Table contain the absolute address and field length for the insertion of a parameter as well as the limitations on the specification of the parameter itself. The second word of the Symbolic Parameter Entries (SPE) contain the absolute equivalents of the parameters expressed symbolically on the Parameter Cards.

C. Kinds of Parameters

There are three kinds of parameters which may be specified: numeric parameters, character parameters, and symbolic parameters.

1. numeric - a. signed or unsigned, whole or fractional numbers expressed in binary, octal, or decimal form which are to be converted to unsigned VFL binary numbers or signed VFL binary numbers of byte size one with field length from 1 to 64.

   b. signed decimal whole numbers which are to be converted into the internal 32-bit "integer" form as required by Alpha.

   c. decimal floating-point numbers which are to be converted into internal 64-bit binary floating point numbers.

2. character - characters which are to be converted from

BCD to Harvest Character Set (HCS) coding (See Section III.C.2).

3. symbolic - the name for an absolute parameter; the absolute parameter itself is contained in a Symbolic Parameter Entry (SPE) of the Symbol Translator Table.

If the kind of parameter is numeric or character, the Parameter Location Entry (PLE) specifies the type of parameter which can be used and the limitations on it; if the kind of parameter is symbolic, the type and limitation fields specified in the entry are disregarded except for those limiting the total length of the resulting binary parameter.

II. Pseudo-Op and Symbol Translator Table Formats

A. Parameter Location Entry (PLE)

1. Pseudo-Op Format

```
                                      #-OF-NOS.
PARAM |
IDENT | PLE, LOC, TYPE(FRL),            or      (EX/MAX, FLD-LNG, MUST)
      |                               #-OF-CHAR
```

a. Parameter Identification - the symbolic name for the entry in the Symbol Translator Table as well as the parameter identification to be specified on a Parameter Card.

b. PLE - the pseudo-op.

c. Symbolic Location - the name of the memory area into which the parameter is to be inserted.

d. Type - the type of parameter to be specified on the Parameter Card; if a numeric parameter, it indicates whether it is a signed or unsigned VFL parameter to be expressed in binary, octal, or decimal form (signed: BS, OS, DS; unsigned: BU, OU, DU), an "integer" (I), or a floating-point parameter (F); H indicates a character parameter; S indicates a symbolic parameter.

e. Fraction Length - if a fractional numeric parameter is to be specified, it indicates the number of fractional bits in the resulting VFL binary number; the fraction length is expressed in decimal form.

f. (1) Number of Numbers - when a numeric parameter is specified, it indicates an exact or maximum number of numbers which must or can be specified on the Parameter Card for this parameter.

(2) Number of Characters - when a character parameter is specified, it indicates the exact or maximum number of characters which must or can be accepted by the program for this parameter.

The field is expressed in decimal form with a maximum value of 262,144 (32,768 words of 8-bit characters).

Note - If the type of parameter is symbolic, the combination of the Number of Numbers and Field Length fields is used to limit the number of bits in the resulting number.

g. Exact/Maximum - this field indicates whether the number of numbers or number of characters field contains an exact number of numbers or characters which the program must have for this parameter or whether the number is the maximum number of numbers or characters which the program can accept. EX indicates exact, MAX indicates maximum.

h. Field Length - when a VFL numeric parameter is to be specified, this field indicates the field length in bits of the resulting binary number. The field is expressed in decimal form with a maximum acceptable value of 64.

i. Must - this field indicates whether or not this parameter has to be inserted in the program for it to function properly (i.e., that it must be specified on Parameter Cards accompanying the request for the program). MUST indicates that the parameter has to be specified; a blank field indicates that the parameter need not be specified each time the program is run.

*Note - The Fraction Length, Field Length, and Must fields can be omitted if not used, e.g., the format could be

```
                                #-OF-NOS.
PARAM |                            or      (EX/MAX)
IDENT | PLE, LOC, TYPE,         #-OF-CHAR
```

The other fields must be specified.

2. Symbol Translator Table Entry Format

```
0   0   0   1   1   2   2   2   3   3   4   4   4   5   5   6
0---4---8---2---6---0---4---8---2---6---0---4---8---2---6---0---
```

PARAMETER IDENTIFICATION								
INSERTION ADDRESS 24	TYPE 4	FRACTION LENGTH 7	# OF NUMBERS # OF CHARACTERS 18	EX MAX	FIELD LENGTH 7	MUST	ERROR	INPUT

a. Parameter Identification - the parameter identification of the PLE pseudo-op in HCS coding; for a name of less than eight characters, the word contains HCS blanks in the rightmost character positions.

b. Insertion Address - the starting bit address of the memory area into which the parameter is to be inserted.

c. Type - Ø: Null, 1: Binary, 2: Octal, 3: Decimal, 4: Character, 5: Integer, 6: Floating-Point, 7: Symbolic.

d. Fraction Length - Ø: no fraction length specified, 1-64: valid fraction length for numeric parameters, 65-127: invalid.

e. Number of Numbers or Number of Characters - Ø: indicates a value of 262,144, 1-262,143: actual value.

f. Exact/Maximum - Ø: Maximum, 1: Exact.

g. Field Length - Ø, 65-127: Invalid; 1-64: actual value.

h. Must - Ø: Not Must, 1: Must.

i. Error - An indicator bit that can be set by the problem program at parameter check time to indicate that the reason for its rejection of the specified parameters is associated with this parameter. Ø: No Error, 1: Error.

j. Input - An indicator bit that is set to 1 by the Production Program File Generator when the parameter is inserted into the program. Ø: Not Input, 1: Input.

B. Symbolic Parameter Entry (SPE)

1. Pseudo-Op Format

```
SYMBOLIC   |
PARAMETER  | SPE, PARAMETER(LENGTH)
NAME       |
```

a. Symbolic Parameter Name - the symbolic name for the entry in the Symbol Translator Table as well as the parameter name to be specified on a Parameter Card.

b. SPE - the pseudo-op.

c. Parameter - the parameter or symbol for the parameter; treated by the assembly as an address field except that the parameter can be from 1 to 55 bits long.

d. Length - the length in bits of the parameter, expressed in decimal form with a maximum acceptable value of 55.

2. Symbol Translator Table Entry Format

```
0   0   0   1   1   2   2   2   3   3   4   4   4   5   5   6
0---4---8---2---6---0---4---8---2---6---0---4---8---2---6---0---
+--------------------------------------------------------------+
|                   SYMBOLIC PARAMETER NAME                    |
+-------------------------------------------------------+------+-+
|                 PARAMETER                             |LENGTH|Ø|
|                                                     55|     6|3|
+-------------------------------------------------------+------+-+
```

a. Symbolic Parameter Name - the name for the parameter in HCS coding; for a name of less than eight characters, the word contains HCS blanks in the rightmost character positions.

b. Parameter - the absolute parameter, right justified, i.e., the low-order bit of the parameter is in bit position 54.

c. Length - the length in bits of the parameter.

*Note - bits 61, 62, 63 of the second word are zero; they correspond to the MUST, ERROR, INPUT bits of the Parameter Location Entry (PLE).

III. Parameter Specification

Problem program parameters are specified in columns 12-80 of Parameter Cards which accompany the request for each program. These parameters are expressed by a parameter identification and an absolute parameter or a symbolic name for a parameter.

A. General Rules

1. The parameter identification is separated from its associated parameter by an equals (=) character (PARAM IDENT = PARAMETER).

2. When there is space. and it is desired, more than one parameter (parameter identification plus parameter) can be punched on one card. These parameters are separated by a question mark (12-4-8 punch).

3. When more than one card is required to specify a parameter, the parameter identification is repeated on a second card and is followed by a +1, that of the third card by +2, etc. (Card 1: JOE = PARAM, Card 2: JOE + 1 = PARAM, etc.).

4. All blanks in the parameter specification field of a Parameter Card, except for those in columns being entered as character parameters, are ignored.

5. A parameter identification or symbolic name for a parameter can be any combination of no more than eight alpha-numeric (A-Z, Ø-9) characters of which the first is alphabetic (A-Z).

B. Numeric Parameters

1. General Rules

a. A numeric parameter can consist of one or more numbers, each expresses in the same form (signed or unsigned binary, octal, or decimal VFL; integer; floating point). The maximum number of numbers which can be specified or the exact number of numbers which must be specified are restricted by the program in the Parameter Location Entry (PLE) for the parameter.

b. When more than one number is specified, the numbers are separated by a comma (0-4-8 punch).

2. VFL Parameters

   a. All VFL parameters (binary, octal, decimal) will have the resulting field length and fraction length specified in the Parameter Location Entry (PLE) in the Symbol Translator Table of the problem program.

   b. All the numbers must be either signed or unsigned and not a mixture of both. A signed number may be preceded by a plus or minus sign; the lack of an explicit sign indicates a positive number.

3. "Integer" Parameters

   An integer will consist of a minus sign or plus sign (which may be omitted if the number is positive) and from 1 to 9 decimal digits whose value is less than 536, 870, 912 (2^{29}).

4. Floating-Point Numbers

   a. A floating-point parameter will be expressed in decimal as a decimal floating-point number in the following form:

      Mixed Fraction Part:

         Sign of Fraction - a plus or minus sign (may be omitted if the fraction is positive).

         Whole Number - an integer between 1 and 9.

         Decimal Point

         Fraction Part - from 1 to 15 decimal digits.

      Exponent Part:

         Exponent Indicator: "E"

         Sign of Exponent - a plus or minus sign (may be omitted if the exponent is positive).

Exponent Value - from 1 to 3 decimal digits from 0 to 308.

b. If the exponent is zero, the exponent part may be omitted.

c. Examples -

-2.35×10^5	:	-2.35E5
+2.35	:	2.35
.01	:	1.0E-2

C. Character Parameters

1. A character parameter must begin in the card column immediately following the equals sign which separates it from the parameter identification.

2. Until such time as 120-character (including blank) card punches become available, the monome-dinome method of representing 120 characters with the 49-character set (Type 9) will be used. This method is as follows:

Two symbols in the 49-character set are reserved as indicators for characters outside of the 49-character set range. The two symbols reserved are the asterisk (*) and the number sign (#). Any other character within the 49 will be punched in a single card column as has been done in the past. Thus, we can represent 47 (including blank) of the 49 characters in the usual manner. Each remaining character of the 120-character Harvest Character Set will be represented by punching two of the available 49 characters as a dinome. The first of the pair of characters will be either the asterisk (*) symbol or the number sign (#) symbol. The second character of the dinome may be one of the normal alphabetic or numeric (A-Z, Ø-9) or an equals (=)

character. The asterisk itself will be represented as the dinome *= and in like manner, the number sign as #=.

3. A character parameter can occupy as many cards as needed so long as the number of characters is the exact number or does not exceed the maximum number specified by the program in the Parameter Location Entry for the parameter. (The maximum allowable number is 262,144 characters.)

4. If a full card is not used in the punching of characters, the last valid character is followed by a ** dinome (which is not one of representations for the 120-character HCS).

D. Symbolic Parameters

1. When a symbolic parameter name is used, the letter S enclosed in parens is punched on the Parameter Card following the parameter identification (PARAM IDENT(S) = PARAMETER).

2. A symbolic parameter can contain more than one symbolic parameter name. The total length in bits of the resulting absolute parameter cannot differ from the exact or exceed the maximum indicated in the associated Parameter Location Entry; this number of bits is indicated by the Field Length and Number of Numbers field for numeric and symbolic parameter entries and by the Number of Characters field for character parameter entries.

3. A symbolic parameter name can consist of any combination of no more than eight alpha-numeric (A-Z, Ø-9) characters of which the first is alphabetic (A-Z).

E. HOPS Problem Program Parameters and Data File Descriptors

1. Problem programs requiring parameters concerning the location of data within an input record, and possibly the form of said data (e.g., floating-point number, integer, etc.), require the location to be specified in numeric terms of increment and length and the data form, if needed, by a letter (e.g., "F" or "I"); the increment, length, and data form being separated by commas. Therefore, when data descriptors are employed in parameter statements, they must be replaced by the type of information which the program needs, before the parameters can be used by the program. Consequently, a descriptor "name" in a parameter statement will in effect be replaced by the location information extracted from the data descriptor. Not all programs of the type being discussed need the data form from the descriptor; so, for those that do, the descriptor name must be accompanied by an indicator. A plus (+) sign following the descriptor name will cause the form indicator to be extracted from the descriptor along with the increment and length information.

2. Each descriptor entry that appears in a parameter statement will be preceded by a special"begin" symbol (*#) and followed by a special "end" symbol (#*). These symbols are not valid HCS configurations and as such do not impose any limitation on the use of valid ones in the rest of the statement. Thus, when used in a parameter statement, a descriptor will start with the special begin symbol, then the descriptor name for the data, followed by a plus sign if needed, and will end with

the special end symbol.

3. Parameters employing descriptors are in the character parameter class. The descriptors in a parameter statement are clearly marked off by the special symbols and may be interspersed among other character parameters in the statement as needed by the program. One more item is necessary when using descriptors: they must be specifically related to a particular data file. This is accomplished by following the parameter identification (or name) with the appropriate program file symbol and separating the two with a slash (/). The parameter statement will then appear as follows:

```
PARAM IDENT/FILE SYMBOL = ...(begin symbol) NAME + (end symbol)...
PARAM IDENT/FILE SYMBOL = ...*# NAME + #*...
```