Published: 01/16/67

Identification

procedures to obtain option stack status
option_frameno, option_names, option_values
C. Marceau

Purpose

These option procedures return information about the current status of the options stack. Option_frameno returns the number of the current frame of the options stack. Option_names returns the names of options which are set. Option_values returns the values (in all frames) of a given option.

Usage

```
call option_frameno(n);
```

n--the number of the current stack frame, returned by option_frameno. The calling program should include the declaration

```
dcl   n   fixed;
```

To obtain the name of the mth option

```
call option_names(m,name);
```

The "first option" means the first option to be found by scanning the hash table, the mth option is the mth one to be found. The ordering of options by this scheme is not alphabetic; by the very nature of a hash table, the ordering is pseudo-random. To obtain the names of all options which are set, call option_names for m = 1, 2, 3, and so on. Option_names returns names of local options, global options, and local values of global options. Local values of global options are given in the form

```
"caller.name"
```

where "name" is the name of the global option, and "caller" is the name of the procedure for which the option is locally set. When option_names is called for some m greater than the number of options which are set, it signals an error condition:

```
signal condition(options_401);
```

Option_names expects the following declarations:

```
dcl   m   fixed,

      name char  (K);
```

where $0 < K \leq 64$. K should be at least as long as the name of the longest option the user has set.

To obtain the values of an option in all frames,

```
call option_values(name,sws,specs,n);
```

name--name of the option.
sws--an array. sws(i) is the switch for name in frame i.
specs--specs(i) is the specification for name in frame i.
n--the number of the frame in which name was first set.

The calling procedure should contain the following declarations:

```
dcl   name  char (K),
      sws (M)  bit (1),
      spec (M) char (L) var,
      n  fixed;
```

where M is some number $\geq$ the current frame number. The current frame number may be obtained by calling option_frameno (q.v.). $0 < K \leq 64$, $0 \leq L \leq 512$.

Implementation

The implementation of option_frameno and option_names follows trivially from the representation of the options stack as described in BX.12.01. Option_frameno returns the value of option_seg.fno (an element of the option_seg structure--see BX.12.01).

Option_names peruses the hash table for non-vacant entries. When it finds the mth non-vacant entry (i.e., a relative pointer to an option header) it picks up the name of the option from the header.

Option_values obtains its first value--sws(n) and specs(n)--for the option from the option header in frame n. Then from the next setting of the option, in frame m, the procedure obtains the values of sws(m) and specs(m). For all j such that $n < j < m$, sws(j) = sws(n) and specs(j) = specs(n). Option_names then goes on to the next setting, and so on to the current setting of the option in frame q. Then if k = the current frame number, sws(j) = sws(q) and

specs(j) = specs(q) for $q < j \leq k$.

For all $j < n$ and $j > k$

sws(j) = "0"b
specs(j) = "" (null specification)