

Published: 1/6/67

Identification

procedures to check options
read_opt, read_global
C. Marceau

Purpose

The read_opt primitive checks local options, i.e. options which affect one procedure only. Read_global checks global options, i.e. options which affect more than one procedure. Read_global checks first to see if the option is set locally for the calling procedure; then it checks the global value. Read_global should be used to check any option which has meaning for more than one procedure. Section BX.12.00 contains an overview of options, including a discussion of local and global options.

Usage

To check the value of the option name in frame n,

```
call read_opt (name, n, switch, spec, set);
```

If $n = 0$, read_opt returns the value of name in the current frame.

```
switch = "0"b if the option is set OFF,  
        = "1"b if the option is set ON
```

```
spec - the specification of the option
```

```
set = "0"b if the option is not set  
     = "1"b if the option is set
```

If $n < 0$ or $n > k$, where k is the number of the current frame, read_opt signals an error:

```
signal condition (options_101);
```

If name is unset, read_opt returns the following values:

```
switch = "0"b
```

```
spec = "" (null character string)
```

```
set = "0"b
```

The declarations of the arguments to read_opt are

```
    dcl name char (J),
        n fixed,
        (switch, set) bit (1),
        spec char (L) var;
```

where $0 < J \leq 64$ and $0 \leq L \leq 512$.

To check the value of a global option:

```
    call read_global (caller, name, n, switch, spec, set);
```

caller - the name of the calling procedure. Other arguments are the same as for read_opt. Caller is declared by:

```
    dcl caller char (M);
```

where M is any integer such that $0 < M \leq 63 - Q$, where Q is the length of the longest option name checked by caller.

If $n < 0$ or $n > k$, where k is the current frame number, read_global signals an error:

```
    signal condition (options_102);
```

Read_global first checks to see if name is set locally for caller, i.e. if "caller.name" is set. (This convention for local settings of global options is explained in BX.12.00.) If name is not set locally for caller, read_global checks for a global value of name.

If name is set neither locally nor globally, read_global returns the following values:

```
switch = "0"b
```

```
spec = "" (null character string)
```

```
set = "0"b
```

If name is set either locally or globally, read_global returns

```
set = "1"b
```

and the switch and specification of the local setting.

Implementation

The implementation of `read_opt` follows trivially from the representation of options described in BX.12.01. `read_opt` hashes the option name to get to the option header. The option header points directly to the current setting for the option. To find the value of the option in frame `n`, `read_opt` goes back along the chain of settings of the option until it finds a setting in some frame $m \leq n$. The value of the option in frame `m` is valid also in frame `n`.

`Read_global` calls `read_opt` for the value of "caller.name". If `caller.name` is unset, `read_global` calls `read_opt` for the value of "name".