Identification
procedures to set options
modset, modopt
C. Marceau

Purpose
Modset sets an option in a specified frame. Modopt sets the option in that frame and all subsequent frames. Suppose the current frame is frame 5 and modset is called to set option "zilch" in frame 2. Modset changes the value of "zilch" in frame 2 but does not alter the value of zilch in frames 3, 4, or 5. If modopt is called, it changes zilch in frame 2 as well as in frames 3, 4, and 5.

Section BX.12.00 presents an overview of options and of the options stack.

Usage

    call modset(name, n, switch, spec);
    call modopt(name, n, switch, spec);

name--name of the option to be set
n--frame in which it should be set
switch = "1"b if the option is to be set on,
      = "0"b if the option is to be set off
spec--specification for the name.

The arguments to modset (or modopt) should have the following declarations:

    dcl name char (K),
    n fixed,
    switch bit (1),
    spec char (L) var;

where 0 < K ≤ 64 and 0 ≤ L ≤ 512.
If \( n < 1 \) or \( n > k \), where \( k \) is the number of the current frame, modset (or modopt) signals an error:

\[
\text{signal condition (options_301)};
\]

**Example:**

The following example shows how the user might set and read options. In this example, modopt is used in conjunction with read_opt and read_global (BY.9.01), and push_opt and pop_opt (BY.9.02).

The user wishes to set certain options for the duration of his group of procedures \( \text{phi, psi, and chi} \). "Help" is an option which all three check. In addition, chi should run with the "alone" option off whenever it runs with \( \text{phi and psi} \) ("alone" is on only when chi runs alone).

On entering \( \text{phi} \):

\[
\text{phi: proc(arg1, arg2);} \\
\text{call push_opt;} \\
\text{call modopt(help, "1"b, "", 0);} \\
\text{call modopt(alone, "0"b, "", 0);} \\
\]

Before returning from \( \text{phi} \):

\[
\text{call pop_opt(0);} \\
\text{return;} \\
\text{end;} \\
\]

In this example the first call to modopt sets help in the current frame. The setting is on ("1"b), with no specification ("" is a null character string). The second call to modopt sets alone off ("0"b), with no specification, in the current frame.

Chi checks the "alone" option by calling

\[
\text{call read_opt("alone",0,switch,spec,set);} \\
\]
This call returns the value of "alone" in the current frame (second argument is 0.) Switch = "1"b if alone is on, and = "0"b is alone is off. Spec is the specification of alone, if any. "Set" = "1"b if alone is set, and = "0"b if alone is unset. In this example read_opt returns

```
switch = "0"b
spec = ""
set = "1"b
```

Phi checks the help option:
```
call read_global(phi, "help", 0, switch, spec, set);
```

Psi calls
```
call read_global(psi, "help", 0, switch, spec, set);
```

Chi calls
```
call read_global(chi, "help", 0, switch, spec, set);
```

Read_global returns to all of them:
```
switch = "1"b
spec = ""
set = "1"b
```

Implementation

Whenever modset (or modopt) is called to set an option which is currently unset, modset (or modopt) calls
```
call addopt(name, n, switch, spec);
```
to create a header (see BX.12.01) for the option. Addopt is described in BY.9.05.

When modset is called to set name in frame n, and name is already set (i.e., has a header in some frame), then modset modifies (or creates) the setting for name in frame n. (See BX.12.01 for the representation of options and definitions of header, setting, etc.)

Modset must further ensure that the value of name in frame n+1 remains unchanged. (If the option is unset in frame n+1, the value in frame n holds also for frame n+1.) Therefore modset must create a setting for name in frame n+1, if none already exists.
Further, the header for an option must lie in a lower frame than any other setting. If the header for \textit{name} is in frame \( m > n \), then modset must replace the header in frame \( m \) with a setting, and make a header for name in frame \( n \). If \( m > n = 1 \), then modset also creates a header for \textit{name} in \texttt{perm_op_list}.

When \texttt{modopt} is called to set \textit{name} in frame \( n \) and \textit{name} is unset, \texttt{modopt} calls \texttt{addopt} to create a header for \textit{name}. If \textit{name} is already set, \texttt{modopt} modifies (or creates) the setting of \textit{name} in frame \( n \), then deletes all subsequent settings so that the value of \textit{name} in frame \( n \) is valid in all subsequent frames.

If \textit{name} has a header in frame \( m > n \), \texttt{modopt} creates a header for \textit{name} in frame \( n \) (the header in frame \( m \) is deleted along with all other settings). If \( m > n = 1 \), \texttt{modopt} creates a header for \textit{name} in \texttt{perm_op_list}.

\texttt{Modset} and \texttt{modopt} allocate settings and specifications in an area (\texttt{option_seg.space}—see BX.12.01) in the options stack. If the PL/I \texttt{area} condition is signalled as the result of an attempted allocation, \texttt{option_seg.space} must be expanded. On \texttt{area}, \texttt{modset} (or \texttt{modopt}) calls \texttt{area$_\$redef} to double the size of \texttt{option_seg.space}.