Identification
procedure to check options of another user
read_user_opt

C. Marceau

Purpose
It is sometimes necessary for a process belonging to one user (or a
daemon) to check the options of another user. One example is provided by
the no_mail option. Before sending mail to a user, the mail command must
check whether the user accepts mail, i.e., must check the status of his
"no_mail" option.

Read_user_opt is similar to read_opt. There is no facility for checking
global options belonging to another user.

Usage

call read_user_opt
   (person, projid, name, switch, spec, set)

read_user_opt checks the name option in the permanent options list of
the user specified by person and projid.

person—the name of the user as he is known to the system (the name or
unique mnemonic by which he logs in).
projid—id of the project which person works on.
name-name of the option to be read.

switch-read_user_opt returns switch = "l"b if the option is on, "0"b if the option is off.

spec-specification of the option (if any)

set="0"b if name is unset, = "1"b if name is set.

The calling procedure should contain the following declarations:

```
dcl person char (31) var,
    projid char (31) var,
    name char (K),
    switch bit (1),
    spec char (L) var,
    set bit (1);
```

where $0 < K \leq 64$, and $0 \leq L \leq 512$.

If the option is unset, read_user_opt returns

```
switch = "0"b
spec = "" (null character string)
set = "0"b
```

Access to the user's perm_op_list is controlled by the file system's access control module (see BG.9.00). Normally a user allows anyone to read his perm_op_list segment from the administrative ring. (Read_user_opt
is in the administrative ring.) However, a user can deny any other user
access to his permanent options list (see BX.8.02). It may be that the
user calling read_user_opt is not privileged to read the perm_op_list
segment he specified. In that case read_user_opt signals condition
(options_601). To the permanent options list of the user specified by
person and projid, read_user_opt calls

\[
generate_ptr,
generate_special(pathname, ptr)
\]

which returns a pointer to the desired segment. The pathname is of the
form:

\[>user_profile_dir>person.projid>perm_op_list\]

If the user can access the segment, read_user_opt treats the segment as
a controlled PL/I structure, option_seg, described in BX.12.01. I.e.,

\[ptr->option_seg\]

refers to the permanent options list from which values are to be read.

If the user may not access the segment, read_user_opt signals an error:

\[signal condition (options_601)\]

Read_user_opt hashes the option name to find the header for name. The
header points to the current setting of name. Read_user_opt goes back
along the chain of settings of name until it finds a setting in frame m ≤ n. The value in frame m is valid in frame n.