Identification

Printerr - a procedure to format and print error comments contained in a user's error segment
D. Widrig, K.J. Martin

Purpose

The printerr procedure gives the user a convenient way of printing selected parts of his error segment. It is expected that calls to printerr will be imbedded at critical places in a user's program so as to advise the user of the occurrence of certain errors. It is also expected that printerr will be useful at command level to obtain a printed record of the error segment.

Usage

call printerr (select);

dcl select char (*) varying;

/* If select = "*", all error descriptions in the segment are printed. If select = "1", only the most recent is printed, if select = "2", only the next most recent, if select = "3", only the third most recent, etc. */

Implementation

The initial implementation of printerr prints all available information of an error description. Printerr calls geterr_complete (BY.11.02) with the skip-if-deleted argument set off ("0"b). The value of n is determined by the value of select. If select is "1", "2", "3", ... then n = select and one call to geterr_complete is enough. If select = "*", geterr_complete must be called first with n = 1, then with n = 2, and so on until geterr_complete reports that the error_out segment is empty.

Following a call to geterr_complete, printerr formats the information and places it in the output stream, "user_output". Formatting is accomplished as follows:

dcl break_char char (1), ctl_char\n char (1);

/* described in BY.8.01 */
break_char = ctrl_char\n;

error_line = time||break_char||date||break_char||call_loc||
   break_char||error_loc||break_char||error_code||
   break_char||error_info||break_char||extra_char_info||
   break_char;

The bit string data, extra_bit_info, is converted to a
can character string of octal numbers by repeated applications
of bin_oct (BY.7.01) which operates on only 36 bits of
input. Bin_oct is called as often as necessary to convert
the entire bit string, then the 12-character parts are
concatenated to one string, oct_info. Formatting is continued:

   error_line = error_line||oct_info||break_char;

The argument, attempted_delete, is checked. If it is
"1"b the following statement is executed:

   error_line = error_line||"attempted delete"||break_char;

Printerr then calls write_out (BY.4.02) to write error_line
in the stream, "user_output".

Note that in the initial implementation no attempt is
made to "pretty" up the output line. It is expected that
future refinements will allow the user to have some control
over the formatting of the output line. It is also expected
that future refinements will allow the user additional
selection in what is printed.