MCR10172
Fix accounts_overseer_ to Better Deal with Long Command Lines
Revision 1.1

Eric Swenson

September 10, 2026

1 Problem Description

Ticket https://multics-trac.swenson.org/ticket/321 describes a problem where a long command line to the `accounts_overseer_` results in execution of a truncated command line. It suggests giving the program a larger command buffer and rejecting the input line as too long if it exeeds the buffer length.

`accounts_overseer_` is the process overseer used by the Accounts Administrator (SA1.SysAdmin). While it is not necessary to use this process overseer, as any user on the SysAdmin project can execute the accounting commands, it is nevertheless a good idea to fix this overseer.

Commands are read by `iox_$get_line` and the input buffer provided is only 256 characters long. But more importantly, since `iox_$get_line` returns a zero status code when the input provided is longer than the buffer provided in the case, an error is not reported and a truncated input line is returned. It is, in general non-trivial to determine whether truncation occurs, but this MCR proposes a workaround.

2 Proposed Fix

The proposed fix includes making the input buffer based on a temp segment, created with `get_temp_segment_` and released with `release_temp_segment_`. This effectively makes the buffer 1,044,480 characters long.

It also proposes checking to see if the number of characters read is equal to the input buffer length and whether the last character is a newline. If the maximum command line length is returned and the last character is not a newline, an error is signalled and another command line read. While this additional check will probably never be necessary, it is added because otherwise, the command line would still be executed truncated if the command line were greater than 1,044,480 characters long, which it could be (in theory) if the user had 256K segments enabled.

In addition, since `com_err_` is called with a literal string many times in the program, a manifest constant (ME) is added to allow the use of a constant string.

```
cpa [lpn accounts_overseer_.pl1] ==

A22        1) change(87-08-25,GDixon), approve(88-08-15,MCR7969),
A23           audit(88-08-04,Lippard), install(88-08-29,MR12.2-1093):
Changed by B to:
B22        1) change(1987-08-25,GDixon), approve(1988-08-15,MCR7969),
B23           audit(1988-08-04,Lippard), install(1988-08-29,MR12.2-1093):


Inserted in B:
B27        2) change(2026-09-10,Swenson), approve(2026-09-10,MCR10172):
B28           Updated to use a temp segment for the command line buffer rather than a
B29           256-character buffer to support longer command lines.
Preceding:
A27                                                    END HISTORY COMMENTS */


A34      dcl  input_line char (224) defined (whole_line) pos (33);
Changed by B to:
B37      dcl  input_line char (CHARS_PER_SEGMENT - 32) defined (whole_line) pos (33);


A42      dcl  whole_line char (256);
Changed by B to:
B45      dcl  temp_seg_ptr ptr;
B46      dcl  whole_line char (CHARS_PER_SEGMENT) based (temp_seg_ptr);


Inserted in B:
B62      dcl  get_temp_segment_ entry (char(*), ptr, fixed bin(35));
Preceding:
A58      dcl  ioa_ entry options (variable);


Inserted in B:
B70      dcl  release_temp_segment_ entry (char(*), ptr, fixed bin(35));
Preceding:
A65      dcl  trace_stack_ entry (ptr, fixed bin, fixed bin, char (32) aligned);


Inserted in B:
B88      dcl  ME char(18) int static options (constant) init ("accounts_overseer_");
Preceding:
A82      dcl  NL char (1) aligned int static init ("

Inserted in B:
B99               call setup(code);
B100              if code ^= 0 then do;
B101                   call com_err_ (code, ME, "Unable to initialize process overseer.");
B102                   return;
B103              end;
```

```
B104
Preceding:
A92               call cu_$get_cl_intermediary (cl_safe_intermediary);


A102                      call com_err_ (code, "accounts_overseer_", "");
A103                  end;                                    /* don't enable quits in
    absentee, abs_io_ complains */
Changed by B to:
B115                      call com_err_ (code, ME, "");
B116              end;                                        /* don't enable quits in
    absentee, abs_io_ complains */


A109                      call com_err_ (code, "accounts_overseer_", "Unable to set
    working directory ^a.", path);
Changed by B to:
B122                      call clean_up();
B123                      call com_err_ (code, ME, "Unable to set working directory ^a.",
    path);


A116              if code ^= 0 then call com_err_ (code, "accounts_overseer_", "");
Changed by B to:
B130              if code ^= 0 then call com_err_ (code, ME, "");


A121              if code ^= 0 then call com_err_ (code, "accounts_overseer_", "");
A122              if ll <= 1 then go to restart1;
Changed by B to:
B135              if code ^= 0 then call com_err_ (code, ME, "");
B136              if ll <= 1 then go to restart1;
B137              /* Check to see if our input line was truncated. */
B138              if ll = length(input_line) & substr(input_line,ll,1) ^= NL then do;
B139                  call com_err_ (0, ME, "Command line too long. Maximum command line
    is ^d", ll-1);
B140                  goto restart;
B141              end;
B142


A138                      if code ^= 0 then call com_err_ (code, "accounts_overseer_",
    "");
Changed by B to:
B158                      if code ^= 0 then call com_err_ (code, ME, "");

Inserted in B:
B171                      call clean_up();
Preceding:
A151                      return;
```

```
A157              if code ^= 0 then call com_err_ (code, "accounts_overseer_", "");
Changed by B to:
B178              if code ^= 0 then call com_err_ (code, ME, "");


Inserted in B:
B186              call setup(code);
B187              if code ^= 0 then do;
B188                   call com_err_ (code, ME, "Unable to initialize process overseer.");
B189                   return;
B190              end;
Preceding:
A165              go to restart;


A206              if code ^= 0 then call com_err_ (code, "accounts_overseer_", "performing
    resetread");
Changed by B to:
B232              if code ^= 0 then call com_err_ (code, ME, "performing resetread");


A210              if code ^= 0 then call com_err_ (code, "accounts_overseer_", "");
A211              call iox_$get_line (iox_$user_input, addr (input_line), length
    (input_line), ll, code);
A212              if code ^= 0 then call com_err_ (code, "accounts_overseer_", "");
Changed by B to:
B236              if code ^= 0 then call com_err_ (code, ME, "");
B237              call iox_$get_line (iox_$user_input, addr (input_line), length
    (input_line), ll, code);
B238              if code ^= 0 then call com_err_ (code, ME, "");


A234              if code ^= 0 then call com_err_ (code, "accounts_overseer_", "performing
    resetread");
Changed by B to:
B260              if code ^= 0 then call com_err_ (code, ME, "performing resetread");


A239         end;
Changed by B to:
B265     setup: procedure(P_code);
B266
B267         dcl P_code fixed bin (35) parameter;
B268
B269         temp_seg_ptr = null();
B270


A234              if code ^= 0 then call com_err_ (code, "accounts_overseer_", "performing
    resetread");
Changed by B to:
B260              if code ^= 0 then call com_err_ (code, ME, "performing resetread");
```

```
A239          end;
Changed by B to:
B265     setup: procedure(P_code);
B266
B267          dcl P_code fixed bin (35) parameter;
B268
B269          temp_seg_ptr = null();
B270
B271          call get_temp_segment_ (ME, temp_seg_ptr, code);
B272          if code ^= 0 then do;
B273               call com_err_ (code, ME, "Could not allocate temporary segment for
    command line buffer");
B274               P_code = code;
B275               return;
B276          end;
B277
B278          P_code = 0;
B279          return;
B280     end setup;
B281
B282     clean_up: procedure ();
B283         if temp_seg_ptr ^= null then do;
B284              call release_temp_segment_ (ME, temp_seg_ptr, code);
B285              if code ^= 0 then do;
B286                   call com_err_ (code, ME, "Could not release temporary segment");
B287                   return;
B288              end;
B289         end;
B290         return;
B291
B292     end clean_up;
B293
B294     %page;
B295     %include system_constants;
B296
B297     end accounts_overseer_;

Comparison finished: 20 differences, 94 lines.
```

3 Documentation

No changes to the documentation are required as none of this is documented now.

4 Test

The updated accounts overseer has been tested with short and long command lines and no issues were found.

5 Version History

2026-09-07	1.0	Initial version of the MCR.
2026-09-10	1.1	Updated to use temp segment rather than larger automatic buffer.