

MCR10106

Fix Bad Error Messages in generate_mst

Revision 1.1

Eric Swenson

December 22, 2021

1 Problem Description

When the `generate_mst` command encounters errors (for example missing segments), it emits bogus error messages to the output listing segment when the `-list` control argument is used. These error messages look like this:

```
bootload_slm_manager 08/13/93 1647.9 Anthony.SysEng.a alm
bootload_tape_fw 08/13/93 1647.8 Anthony.SysEng.a alm
template_slm_ 08/13/93 1647.7 Anthony.SysEng.a alm
\c \000\000\000\000\000\000\000\000$ \00\000\004\000\000\000\
  \c000\234\400#\006\656\000\000\000\234\400#\006\424\000\000\000\234\400#
\c dd>s\000\234\400#\005h\000\000\000\000\000\000core\0
\c00\247\400#\031\514\000\000\000\247\400#\032:\000\000REWP \000\000\000\000
\c\000\000\000\000?\777\000#\000\001\000\000creation_time_string \00
\c0\000S#\342\663\456\300\000\334\000#
...
\c\000\000\000\000\000\000\000\000\332\400#\000\000\000\000\000\234\400#\005\73
\c0\000\000?\777\000#\000\001\000\000?\777\000#\000\001\000\000\777\777\777\777
\c\777\777\777\777\000\234\400\400\000\000?\777\020\636\777\777\777\777\000:\00
\c0\000\000\000\000:\000^\000\000\014\000\000^\025\552\000\000\000\000\000\
  \c000\247\000\000
```

This ticket for this issue is <https://multics-trac.swenson.org/ticket/19>.

2 Proposed Fix

The problem in `generate_mst` is actually in a helper subroutine, `gm_error_`. This is the subroutine called to report errors to the terminal and to emit error lines to the output listing segment. The original implementation of this subroutine is to use `iox.$put_chars` to emit lines to the listing segment. However, it uses a fixed length of 500, rather than the actual length of the output line.

Presumably for some “efficiency” concerns, the buffer used to emit lines to the output listing segment is declared/allocated in the main program, `generate_mst`, and then passed into `gm_error_`. Rather

than allocate a buffer on the stack in `gm_error_` to use when emitting lines to the listing file, this buffer is passed in. This is a very lame optimization.

The proposed fix removes this buffer – `o_ln` from `generate_mst.pl1`, and changing the declaration of `gm_error_` to no longer supply this buffer. Instead, the proposed fix to `gm_error_` uses `ioa_$ioa_switch` to format the output line for the listing segment. The call to `iox_$put_chars` is removed as a result, and the output line

Another issue with `gm_error_.pl1` is that it uses the variable `error_message` for the `com_err_` call (as it should), but then sets the value to the null string before the code that emits the error line to the listing segment. This effectively “loses” the error message.

Finally, for some errors, `generate_mst` is able to determine the name of the segment that is “in error”. It uses this in the output emitted to the `error_output` I/O switch, but doesn’t use it in the output to the listing segment. In a very likely use case – the specification of a missing segment in the `hardcore.header`, the error message is “Missing Segment.”, but there is no indication of the missing segment name in the output listing segment. The proposed fix includes this segment name – if known – in the output listing segment error message.

The fix to `generate_mst.pl1` is shown below:

```
----- generate_mst.pl1

compare_ascii >ldd>tools>source>bound_gm_.s.archive::generate_mst.pl1 ==

A187      dcl o_ln char (132) aligned;
A188
Deleted by B, preceding:
B195      dcl last_path char (32) aligned;                                /* for setting path_found */

A256      pointer, bit (1) aligned, bit (1) aligned, bit (1) aligned, pointer, pointer);
Changed by B to:
B262      bit (1) aligned, bit (1) aligned, bit (1) aligned, pointer, pointer);

A1325      call gm_error_ (0, seg_name, symp, hdrp, "Cleanup handler invoked.",
      addr (o_ln), "1"b, end_found,
Changed by B to:
B1331      call gm_error_ (0, seg_name, symp, hdrp, "Cleanup handler invoked.",
      "1"b, end_found,

A1341      call gm_error_ (code, seg_name, symp, hdrp, gm_message, addr (o_ln),
A1342      fatal, end_found, "0"b, in_p, gm_output_iocbp);
Changed by B to:
B1347      call gm_error_ (code, seg_name, symp, hdrp, gm_message, fatal,
      end_found, "0"b, in_p,
B1348      gm_output_iocbp);

Comparison finished: 10 differences, 48 lines.
```

The fix to gm_error_.pl1 is shown below:

```
----- gm_error_.pl1

compare_ascii >ldd>tools>source>bound_gm_.s.archive::gm_error_.pl1 ==

A10      gm_error_: proc (code, seg_name, symbol_ptr, header_ptr, arg_error_message,
      line_ptr,
Changed by B to:
B20      gm_error_: proc (code, seg_name, symbol_ptr, header_ptr, arg_error_message,

A20      line_ptr ptr,                                /* pointer to line in
      output file */
Deleted by B, preceding:
B30      fatal_error_occurred bit (1),                /* fatal error flag */

A31      dcl line char (95) based (line_ptr),          /* mask for modifying line
      in output file */
Changed by B to:
B40      dcl line char (150),                          /* buffer for output line
      in output file */

A34
Deleted by B, preceding:
B43      dcl (char_index, char_num) fixed bin (17),    /* for printing current
      header line */

A39
Changed by B to:
B47

Inserted in B:
B60      ioa_$ioa_switch entry() options(variable),
Preceding:
A52      parse_file_$parse_file_cur_line ext entry (fixed bin, fixed bin),

A66      error_message = "";
Deleted by B, preceding:
B75
B76      if seg_name ^= "" then do;

A77      if iocb_ptr ^= null then do;
A78      line_ptr -> line = "*****";                /* blank out line and fill
in stars */
A79      substr (line_ptr -> line, 43) = error_message;
A80      call iox_$put_chars (iocb_ptr, line_ptr, 500, code);
A81      end;
```

Changed by B to:

```
B85         if iocb_ptr != null then
B86             call ioa_$ioa_switch (iocb_ptr, "*****: ^a: ^a^/", error_message,
                seg_name);
B87
B88             error_message = "";
```

Comparison finished: 8 differences, 29 lines.

Note that the above change changes the way that errors are emitted to the output listing segment. Now, errors have the format:

```
*****: <error message>: <segment name>
```

The original code didn't include the segment name, and this is useful information. And the original code put the error message at column 43, which seems unnecessary.

3 Testing

The fixed version of `generate_mst` was run against a `hardcore.header` segment with a bogus (non-existent) segment. In the original version of `generate_mst`, this resulted in garbage error messages in the listing segment. With the fixed version, it doesn't.

In addition, various MSTs were generated with the updated `generate_mst` and the listing segments were manually inspected to make sure that they were identical to the original version.

4 Documentation

There are no documentation changes required by this change.

5 Version History

2021-12-21	1.0	Initial version of the MCR.
2021-12-22	1.1	Updates based on MCR and audit review.