

MCR10189

list_sub_tree Loops Forever on a Damaged Archive

Revision 1.0

Eric Swenson

October 13, 2026

1 Problem Description

[Ticket #75](#) describes an issue where `list_sub_tree -archive (lst -ac)` loops forever on an archive with a format error part way through:

ac_comp	lisp_loop_
ac_comp	lisp_macro_macros_
	Format error encountered in archive segment.
	(reading archive)
ac_comp	bound_lisp_library_.bind
ac_comp	display_lisp_object_segment
...	repeats endlessly

The ticket notes it could not be reproduced later for lack of a broken archive; the cause is clear from the source, however.

1.1 Cause

`list_sub_tree` walks directories and archives with `fs_star_`. `fs_star_`'s archive loop calls `archive.$next_component`, passing the previous component pointer (`compP`, initially null to mean "first component"), and then `ARCHIVE_PATH_error`, which returns at once if the code is 0 and otherwise reports the error and transfers according to the caller's action code. Once components have been returned, the entry type is "archive component", so `ERROR_OK` and `ERROR_REJECT` (the defaults when the caller's error handler wants to carry on) go to `NEXT_ARCH_ENTRY` or `REMOVE_ARCH_ENTRY`, back to the top of the loop.

But `archive.$next_component` sets its component pointer argument to null on entry and leaves it null when it returns an error. So the next call starts again at the first component, reaches the same format error, and so on forever. (With `ERROR_REJECT` it also removes the previous, valid, component from the counts each time round.)

2 Proposed Fix

In `fs_star_`'s archive loop, when `archive.$next_component` returns an error, the error is reported as an error for the archive as a whole (entry type set back to `ENTRY_TYPE_ARCHIVE`) and reading of the archive stops:

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

A596          call ARCHIVE_PATH_error (Pdepth, code, Ppath.dir, sn, "", "reading
          archive");
Changed by B to:
B596          if code ^= 0 then do;                                /* archive_ returns compP =
          null after */
B597          sd.entry_data.type = ENTRY_TYPE_ARCHIVE;            /* an error, so going on
          would restart */
B598          call ARCHIVE_PATH_error (Pdepth, code, Ppath.dir, /* at the first
          component and loop */
B599          sn, "", "reading archive");                          /* forever. Report the
          error for the */
B600          go to FINAL_ARCH_ENTRY;                             /* archive and stop reading
          it.          */
B601          end;

Comparison finished: 1 difference, 7 lines.
```

For the archive entry type, `ARCHIVE_PATH_error` already does the right thing: `ERROR_STOP` and `ERROR_ABORT` leave `fs_star_$select`, anything else goes to `FINAL_ARCH_ENTRY`, which terminates the archive and returns to process the next path. This is the same handling as for an error initiating the archive. The components listed before the error are kept; the component counts are no longer disturbed.

Every program that uses `fs_star_` to expand archives benefits, not only `list_sub_tree`.

3 Documentation

No documentation changes.

4 Test

1. Make a damaged archive: copy a valid archive with several components and overwrite part of the header of a component in the middle (for example the header begin string), keeping the bit count. `archive t` reports a format error.
2. `lst -ac` on the directory containing it lists the components before the damaged one, prints the format error once, and goes on to the following entries. The old `fs_star_` loops.
3. `lst -ac` with `-depth`, component star names (`PATH:*.bind`), and on valid archives: unchanged.

5 Files Changed

<code>bound_fs_util.s.archive</code>	<code>fs_star.pl1</code>
<code>bound_fs_util.archive</code>	<code>fs_star_</code> (recompiled)
<code>>sss</code>	<code>bound_fs_util_</code> (rebound)

6 Version History

2026-10-10	1.0	Initial version of the MCR.
------------	-----	-----------------------------