In the command: `history_comment ds SOURCE_PATH -orig ORIG_PATH new` if hcom finds a problem in the original segment, its error message references the `SOURCE_PATH`. This problem was reported in Multics Change Tickets: [190](https://example.com) and [192](https://example.com) (closed as duplicate of 190).

The following example deals with an error message for a file in the x directory.

```bash
hcom display x>tolts_info.incl.pl1 1
>udd>m>gd>w>hcom190>x>tolts_info.incl.pl1
Comment 1 has null approve and audit values a null install value is required.
```

Attempting to correct this warning, and to add a new history comment to this file, we see the warning message is now preceded by the wrong pathname. The problem is in the x>… version of the file; the warning pathname refers to the y>… version.

```bash
hcom compare y>tolts_info.incl.pl1 -orig x==
>udd>m>gd>w>hcom190>y>tolts_info.incl.pl1
Comment 1 has null approve and audit values a null install value is required.
```

**Proposed Changes**

The `hcom_process_seg_.pl1` source (in `>ldd>tools>source>bound_pnotice_.s.archive`) contains a `comment_parse` internal procedure which parses comments in both `SOURCE_PATH` and `ORIG_PATH` segments.
This procedure is passed one of two identical substructures:

```c
    call comment_parse(addr(src_array), d.seg,
       pathname_$component(d.seg.dir, d.seg.ent, d.seg.comp));
    call comment_parse(addr(orig_array), d.orig_seg,
       pathname_$component(d.orig_seg.dir, d.orig_seg.ent, d.orig_seg.comp));
```

```c
comment_parse:
   proc(Pcom_array, seg, com_path);

dcl       Pcom_array                    ptr,
   1 seg                         aligned like d.seg,
   com_path                      char(*);
```

where d.seg describes SOURCE_PATH; d.orig describes ORIG_PATH. Most of the comment_parse code correctly references its seg argument, which identifies the desired substructure to be parsed. However, several internal procedures of comment_parse reference the outer d.seg substructure, instead of the seg argument of comment_parse. These incorrect references cause the bug.

The repair changes are shown below.

```c
compare_ascii >ldd>tools>source>bound_pnotice_.s.archive::hcom_process_seg_.pl1 ==
```

Inserted in B:

```
B96        16) change(2019-09-23,GDixon), approve(2019-09-23,MCR10067):
B97            Fix bug #190 which causes hcom to display history comment errors in the
B98            -orig PATH as if they appeared in the source PATH.
Preceding:
A96                                                          END HISTORY COMMENTS */
```

```
A586          if d.seg.comp ^= "" & index(d.seg.comp,".incl.") > 0 then
B589          if seg.comp ^= "" & index(seg.comp,".incl.") > 0 then

A589          else if d.seg.ent ^= "" & index(d.seg.ent,".incl.") > 0 then
A590          Sincl = TRUE;
A591          else if d.seg.comp ^= "" & after(d.seg.comp,"." ) = "h" then

B592          else if seg.ent ^= "" & index(seg.ent,".incl.") > 0 then
B593          Sincl = TRUE;
B594          else if seg.comp ^= "" & after(seg.comp,"." ) = "h" then

A594          else if d.seg.ent ^= "" & after(d.seg.ent,"." ) = "h" then
B597          else if seg.ent ^= "" & after(seg.ent,"." ) = "h" then

A627          if d.seg.type = 4 & substr(rest,1,2) ^= "/*" |
A628          d.seg.type ^= 4 & substr(rest,1,length(seg.cmt_bgn)) ^= seg.cmt_bgn then do;

B630          if seg.type = 4 & substr(rest,1,2) ^= "/*" |
B631          seg.type ^= 4 & substr(rest,1,length(seg.cmt_bgn)) ^= seg.cmt_bgn then do;
```
if d.seg.type ^= 5 then do; /* dont add blank lines for compin/runoff files */

if seg.type ^= 5 then do; /* dont add blank lines for compin/runoff files */

if d.seg.type = 4 & substr(comment_line,1,length("/*")) ^= "/* " then
   goto LINE_ERROR;
else if d.seg.type ^= 4 & substr(comment_line,1,length(seg.cmt_bgn)) ^= seg.cmt_bgn then

if seg.type = 4 & substr(comment_line,1,length("/*")) ^= "/* " then
   goto LINE_ERROR;
else if seg.type ^= 4 & substr(comment_line,1,length(seg.cmt_bgn)) ^= seg.cmt_bgn then

pathname$_component(d.seg.dir, d.seg.ent, d.seg.comp));

pathname$_component(seg.dir, seg.ent, seg.comp));
Description:
Fix bug #190: hcom diagnoses errors in -orig PATH as if they appeared in the source path.

Installation_directory: >udd>m>gd>w>hcom190;

Build_script: hcom190.mb;

Bound_obj: bound_pnotice_ IN: tools UPDATE; source: hcom_process_seg_.pl1 REPLACE compiler: pl1 -ot;

Testing
The following output shows how the repaired hcom command displays correct pathnames in such error messages; and demonstrates that the error really lies in the x>tolts_info.incl.pl1 segment.

hcom compare y>tolts_info.incl.pl1 -orig x==
>udd>m>gd>w|hcom190>x>tolts_info.incl.pl1
  Comment 1 has null approve and audit values a null install value is required.

ORIGINAL: >udd>m>gd>w|hcom190>x>tolts_info.incl.pl1
NEW SOURCE: >udd>m>gd>w|hcom190>y>tolts_info.incl.pl1

  Changed in source to:
     to second sct for 500/501 MTAR

  14) change(2019-09-23,GDixon), approve(2019-09-23,MCR10067):
    Test fix to history_comment command.
    Inserted in source at end.

r 10:38 1.187 99
**Documentation**

No documentation changes for this repair.

**Version History**

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Author</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-09-23</td>
<td>1.1</td>
<td>Gary Dixon</td>
<td>Change MCR based upon comments from Eric Swenson, who found this bug.</td>
</tr>
</tbody>
</table>