

Subject: MCR10067, Fix pathname in hcom -orig PATH error messages
 Author: Gary Dixon
 Date: 2019-09-23

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](#) and [192](#) (closed as duplicate of 190).

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

```
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.
>udd>m>gd>w>hcom190>x>tolts_info.incl.pl1:
  1) change(1980-03-24,Fakoury), approve(), audit(), install(1986-12-04,MR12.0-1235):
r 10:24 0.595 48
```

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.

```
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.
ORIGINAL: >udd>m>gd>w>hcom190>x>tolts_info.incl.pl1
NEW SOURCE: >udd>m>gd>w>hcom190>y>tolts_info.incl.pl1
  1) change(1980-03-24,Fakoury), approve(), audit(), install(1986-12-04,MR12.0-1235):
Changed in source to:
  1) change(1980-03-24,Fakoury), approve(), audit(1986-11-25,Martinson),
    install(1986-12-04,MR12.0-1235):
    to add 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 10:26 1.142 99
```

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:

```

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));
...
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.

```
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
Changed by B to:
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
Changed by B to:
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
Changed by B to:
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;
Changed by B to:
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;

```

```

A816          if d.seg.type ^= 5 then do;                /* dont add
blank lines for compin/runoff files */
Changed by B to:
B819          if seg.type ^= 5 then do;                /* dont add
blank lines for compin/runoff files */

```

```

A886          if d.seg.type = 4 & substr(comment_line,1,length("/*")) ^= "/* " then
A887              goto LINE_ERROR;
A888          else if d.seg.type ^= 4 & substr(comment_line,1,length(seg.cmt_bgn)) ^=
seg.cmt_bgn then
Changed by B to:
B889          if seg.type = 4 & substr(comment_line,1,length("/*")) ^= "/* " then
B890              goto LINE_ERROR;
B891          else if seg.type ^= 4 & substr(comment_line,1,length(seg.cmt_bgn)) ^=
seg.cmt_bgn then

```

```

A1031          pathname_$component(d.seg.dir,d.seg.ent,d.seg.comp));
Changed by B to:
B1034          pathname_$component(seg.dir,seg.ent,seg.comp));

```

```

A1041          pathname_$component(d.seg.dir, d.seg.ent, d.seg.comp),
Changed by B to:
B1044          pathname_$component(seg.dir, seg.ent, seg.comp),

```

```

A1044          pathname_$component(d.seg.dir, d.seg.ent, d.seg.comp),
Changed by B to:
B1047          pathname_$component(seg.dir, seg.ent, seg.comp),

```

```

A1050          pathname_$component (d.seg.dir, d.seg.ent, d.seg.comp),
Changed by B to:
B1053          pathname_$component (seg.dir, seg.ent, seg.comp),

```

```

A1055          pathname_$component(d.seg.dir, d.seg.ent, d.seg.comp),
Changed by B to:
B1058          pathname_$component(seg.dir, seg.ent, seg.comp),

```

```

A1065          pathname_$component(d.seg.dir, d.seg.ent, d.seg.comp));
Changed by B to:
B1068          pathname_$component(seg.dir, seg.ent, seg.comp));

```

```

A1069          pathname_$component(d.seg.dir, d.seg.ent, d.seg.comp));
Changed by B to:
B1072          pathname_$component(seg.dir, seg.ent, seg.comp));

```

```

A1090          pathname_$component(d.seg.dir, d.seg.ent, d.seg.comp),
Changed by B to:
B1093          pathname_$component(seg.dir, seg.ent, seg.comp),

```

```

A1093          pathname_$component(d.seg.dir, d.seg.ent, d.seg.comp),
Changed by B to:
B1096          pathname_$component(seg.dir, seg.ent, seg.comp),

```

```

A1099          pathname_$component (d.seg.dir, d.seg.ent, d.seg.comp),

```

```

Changed by B to:
B1102                pathname_$component (seg.dir, seg.ent, seg.comp),

A1104                pathname_$component(d.seg.dir, d.seg.ent, d.seg.comp),
Changed by B to:
B1107                pathname_$component(seg.dir, seg.ent, seg.comp),

A1113                pathname_$component(d.seg.dir, d.seg.ent, d.seg.comp));
Changed by B to:
B1116                pathname_$component(seg.dir, seg.ent, seg.comp));

A1117                pathname_$component(d.seg.dir, d.seg.ent, d.seg.comp));
Changed by B to:
B1120                pathname_$component(seg.dir, seg.ent, seg.comp));

Comparison finished: 20 differences, 51 lines.

```

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
```

```
  1) change(1980-03-24,Fakoury), approve(), audit(), install(1986-12-04,MR12.0-1235):
Changed in source to:
```

```
  1) change(1980-03-24,Fakoury), approve(), audit(1986-11-25,Martinson),
    install(1986-12-04,MR12.0-1235):
    to add 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

Date	Revision	Author	Comment
2019-09-23	1.0	Gary Dixon	Initial description of problem & repair.
2019-09-23	1.1	Gary Dixon	Change MCR based upon comments from Eric Swenson, who found this bug.