Multics enhancement ticket [http://multics-trac.swenson.org/ticket/172](http://multics-trac.swenson.org/ticket/172) suggests changing the history_comment (hcom) command to maintain a change-log in Multics info segments. It is useful to understand the origin of changes to info segs. Tying each changed info segment to its supporting Multics Change Request (MCR) can help developers and users understand motivation for changes, and provide additional details that may not be expressed clearly in a changed info seg.

To implement this change, the history_comment command will be extended to add/update comments in segments ending with an .info suffix. Comments will be added at the end of the segment, and will be displayed only by the “hcom display” command, or by printing the entire segment. The history of changes will be ignored by the help command.

Even without this proposed change, every type of info seg can be divided into several chunks (similar to archive components); each chunk is treated like a separate segment. This chunk feature is used to document separate operations of a complex command (like the archive or history_comment command); or document individual entrypoints of a subroutine; or document some/all of the requests of a subsystem.

Each chunk begins with an :Info: header line

```
:Info: <long_name>: <short_name>: <iso_date>  <title>
```

as described in info_seg.gi.info section “Multiple-entry info segs”. Names given in the :Info: line are those segment names that identify the chunk (without the .info suffix). validate_info_seg (vis) reports errors if those names are missing.

help and vis also support a similar chunk header whose content is ignored by help, and whose <xxx_name> tags are not added to the containing info seg:

```
:Internal: <long_name>: <short_name>: <iso_date>  <title>
```

The change proposed here uses this hidden feature to add a history comment to an info seg. The developer will add the first history comment to an info seg; then manually add an :Info: divider if one is not already present in the info seg; then copy the first history comment in a new internal chunk appended to the end of the info seg.

These steps will be documented in info_seg.gi.info, and referenced in hcom.add.info. Once the first history comment is present, the hcom command can then display, update, and add additional comments without manual action by the developer.

A quick search found no existing info segs using an :Internal: header line. So this proposed use for :Internal: chunks should not impact other Multics programs.
Proposed Changes

help and validate_info_seg commands both support :Internal: chunks already. Only the history_comment command needs a small change to support files having an .info suffix.

Since the :Internal: chunk is not parsed by help, the history comment can use any existing comment style. I propose to use the PL/I comment style, since that is highly unlikely to appear in the visible chunks of the info seg.

The hcom database describing supported file suffixes is pnotice_language_info_.cds. An additional array element will be added to support .info segs.

Target bound seg: >sss>bound_pnotice_

SOURCE FILES:
   cds:       pnotice_language_info_.cds

A >ldd>tools>source>bound_pnotice_s.archive::pnotice_language_info_.cds (original)
B >user_dir_dir>Multics>GDixon>w>hcom>hcom.archive::pnotice_language_info_.cds (new)

A90               3 lang_array (41) like pnotice_language_info.lang_array;
Changed by B to:
B90               3 lang_array (42) like pnotice_language_info.lang_array;

Inserted in B:
B300            lang_info.lang_type(42) = 4;
B301            lang_info.lang_name(42) = "info";          /* INFO */
B302            lang_info.comment_start(42) = "/****^ ";
B303            lang_info.comment_end(42) = "*/";
B304
Preceding:
A300       /* Now set up call to create data base */

Comparison finished: 2 differences, 7 lines.

Testing

The following paragraphs show how a history comment is added to a fictitious hello_world.info segment.

pr hello_world.info -nb
  1  2019-05-29  hello_world, hello
  2
  3 Syntax as a command:  hello_world
  4
  5  Function:  This command prints a test message: "Hello, world!".
  6 r 16:41 0.084 0
vis hello_world

>user_dir_dir>Multics>GDixon>w>hcom>hello_world.info

-----------COMMAND info
LINE: 1  2019-05-29  hello_world, hello
LINE: 3  Syntax as a command
LINE: 6  Function

r 16:41 0.071 2

Before editing, this info seg has no :Info: header lines. Start by adding the history comment to the info seg.

hcom add hello.info

Enter approve value:

Enter summary:
Document new hello_world command.
.

>user_dir_dir>Multics>GDixon>w>hcom>hello.info:
1) change(2019-05-29,GDixon):
   Document new hello_world command.
r 16:52 0.269 3

pr hello_world.info -nb
1
2  /****^  HISTORY COMMENTS:
3   1) change(2019-05-29,GDixon):
4       Document new hello_world command.
5                      END HISTORY COMMENTS */
6
7  2019-05-29  hello_world, hello
8
9  Syntax as a command: hello_world
10
11
12  Function: This command prints a test message: "Hello, world!".
r 16:52 0.192 0

Edit the info seg to make the following changes:

1. Add an :Info: divider to the hello_world header line. This includes <long_name>:<short_name>: tags without the .info suffix. That begins the first chunk of the edited info seg.

2. At the end of the info seg, add two blank lines followed by the header line:
   :Internal: history_comment.gi: <iso_date> history_comment
   where <iso_date> is today's date in iso_date format. This begins the second chunk of the info seg, with a name tag telling validate_info_seg it is a General Information type of info seg (not to expect any particular section titles).

3. Then move the history comment (all lines before the :Info: divider) to the end of the info seg, after the just-added :Internal: header line.

This produces a 2-chunk info seg, with the second chunk containing the history comment which the hcom command can manipulate; and which the help command ignores.
pr hello_world.info -nb
 1 :Info: hello_world: hello: 2019-05-29 hello_world, hello
 2
 3 Syntax as a command: hello_world
 4
 5 Function: This command prints a test message: "Hello, world!".
 6
 7
 8 :Internal: history_comment.gi: 2019-05-28 history_comment
 9
 10 /****^ HISTORY COMMENTS:
 11 1) change(2019-05-29,GDixon):
 12    Document new hello_world command.
 13                      END HISTORY COMMENTS */
 15
r 17:19 0.153 0

vis hello_world

>user_dir_dir>Multics>GDixon>w>hcom>hello_world.info

-----------COMMAND info
LINE: 3   Syntax as a command
LINE: 6   Function

-----------GENERAL info
LINE: 11  /****^ HISTORY COMMENTS
Severity 3. These lines of section exceed 71 chars: 14

r 17:20 0.096 2

help -pn hello_world -all
>user_dir_dir>Multics>GDixon>w>hcom>hello_world.info    (4 lines in info)
2019-05-29 hello_world, hello

Syntax as a command: hello_world

Function: This command prints a test message: "Hello, world!".

r 17:20 0.219 6

hcom display hello_world.info

>user_dir_dir>Multics>GDixon>w>hcom>hello_world.info:
 1) change(2019-05-29,GDixon):
     Document new hello_world command.

r 17:23 0.113 4

hcom add_field hello_world.info -approve MCR10065

>user_dir_dir>Multics>GDixon>w>hcom>hello_world.info:
     Document new hello_world command.

r 17:24 0.579 4

The change was also tested against the hcs_.info subroutine info seg, and against the info_seg.gi.info general information segment.
Documentation

This change is documented completely in a new info seg: info_seg.hcom.info. This info seg is append as a new chunk to info_seg.gi.info segment; and pointed to in the main chunk of info_seg.gi.info and in hcom.add.info.

cpa <vis>vis::info_seg.gi.info info_seg.gi.info

A1  2019-05-28  Info Segment Standards
Changed by B to:

Inserted in B:
B327  Adding a history comment to an info seg: The "Multiple-entry info segs"  
B328  format described above provides a method for adding a history comment  
B329  to the end of an info seg. Type: "help info_seg.hcom" for instructions  
B330  on this feature.
B331
B332
B333
Preceding:
A327  Info naming conventions:

Inserted in B:
B354
B355
B356  :Info: info_seg_hcom.gi: info_seg.history_comment: info_seg.hcom:  
B357  2019-05-29  Adding a History Comment to an Info Seg  
B358
B359  Before following this procedure, use the "history_comment display ..."  
B360  command to determine if the info seg includes a history comment entered  
B361  earlier. If so, then use the history_comment command to change these  
B362  comments.
B363
B364
B365  History comment steps:
B366  Use an extension of the multiple-entry info seg, as follows:
B367
B368  1) Add the first history comment to the info seg, using the  
B369  history_comment (hcom) command.
B370
B371  2) Edit the info seg. The just-added history comment text appears at  
B372  the top of this file. After the END HISTORY COMMENTS line of this  
B373  comment text, the next non-blank line is the header line of the  
B374  info seg.
B375
B376
B377  3) The info seg is already in multiple-entry info seg format if this  
B378  header line begins with an :Info: string. In that case, skip to  
B379  step 4 below.
B380
B381
B382
B383  Otherwise, insert the following before the existing header line:
B384
B385  :Info: <long_name>:{ <short_name>:}...
B386
B387
B388
substituting any names already on the info seg into this string.
Names are given without the .info suffix. Any number of names are  
permitted. If the :Info: portion of the line becomes too long, the  
original header may appear on the next line after the :Info: portion.
For an info seg with names hello_world.info and hello.info, the edited line might appear as:


4) At the very end of the info seg, append the following lines.

<blank_line>
<blank_line>
:Internal: history_comment.gi: <iso_date> history_comment

where <iso_date> is the current date in iso_date format. For example:

:Internal: history_comment.gi: 2019-05-29 history_comment

5) Move the new history comment immediately after the :Internal: line created in step 4. All lines that precede :Info: at the beginning of the info seg are the new history comment.

6) Save changes to the info seg.

The info seg now has its first history comment, which may be displayed or updated by the history_comment command without further editing.

Examples:

The following paragraphs show how a history comment is added to a fictitious hello_world.info segment.

pr hello_world.info -nb
   1 2019-05-29 hello_world, hello
   2
   3 Syntax as a command: hello_world
   4
   5
   6 Function: This command prints a test message: "Hello, world!".

r 16:41 0.084 0

vis hello_world

>user_dir_dir>Multics>GDixon>w>hcom>hello_world.info

--------COMMAND info

LINE: 1 2019-05-29 hello_world, hello
LINE: 3 Syntax as a command
LINE: 6 Function

r 16:41 0.071 2

This info seg does not use :Info: header lines. Start by adding the history comment to the info seg.

hcom add hello.info

Enter approve value:

Enter summary:
Document new hello_world command.

>user_dir_dir>Multics>GDixon>w>hcom>hello.info:

1) change(2019-05-29,GDixon):
   Document new hello_world command.

pr hello_world.info -nb

pr hello_world.info -nb

 Edit this info seg as described under "History comment steps" above.

The modified info seg is shown below.

pr hello_world.info -nb

 help -pn hello_world -all

>user_dir_dir>Multics>GDixon>w>hcom>hello_world.info   (4 lines in
Multics Change Request

2019-05-29 hello_world, hello
Syntax as a command: hello_world
Function: This command prints a test message: "Hello, world!".

hcom display hello_world.info

>user_dir_dir>Multics>GDixon>w>hcom>hello_world.info:
1) change(2019-05-29,GDixon):
   Document new hello_world command.

r 17:20 0.219 6
hcom add_field hello_world.info -approve MCR10065

>user_dir_dir>Multics>GDixon>w>hcom>hello_world.info:
   Document new hello_world command.

r 17:24 0.579 4

:Internal: history_comment.gi:  2019-05-29  history_comment

/****^ HISTORY COMMENTS:
1) change(1985-04-12,HSmith), approve(1985-05-25, MCR2355),
   Increased size of test_array to eliminate subscript error.
2) change(1985-05-28, HSmith), approve(1985-05-29 MCR2356),

Written by Gary Dixon.
Add first history comment, when hcom began supporting info segs.
END HISTORY COMMENTS */

At end

Comparison finished: 3 differences, 202 lines.

The change to hcom.add.info is shown below.

cpa [lpn hcom.add.info -lb info] hcom.add.info

/****^ HISTORY COMMENTS:
1) change(1985-05-12, HSmith), approve(1985-05-25, MCR2355),
   Increased size of test_array to eliminate subscript error.
2) change(1985-05-28, HSmith), approve(1985-05-29 MCR2356),

Changed by B to:
/****^ HISTORY COMMENTS:
1) change(1985-05-12, HSmith), approve(1985-05-25, MCR2355),
   Increased size of test_array to eliminate subscript error.
2) change(1985-05-28, HSmith), approve(1985-05-29 MCR2356),

Inserted in B:
To add the first history comment to an info seg,
type: "help info_seg.hcom".

Arguments:
Inserted in B:
B1168
B1169
B1170     :Internal: history_comment.gi: 2019-05-29 history_comment
B1171
B1172     /****^ HISTORY COMMENTS:
B1173       1) change(1988-03-10,LJAdams), approve(), audit(), install():
B1174          Create info seg with full documentation for the hcom command.
B1176          Change hcom.add.info to reference info_seg.history_comment.info for
B1177          instruction on adding the first history comment to an info seg. Also,
add
B1178          a history_comments chunk to end of this info seg.
B1179     END HISTORY COMMENTS */
B1180
B1181
At end

Comparison finished: 3 differences, 28 lines.

Version History

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Author</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-01-08</td>
<td>1.0</td>
<td>Gary Dixon</td>
<td>First version of proposed change to hcom.</td>
</tr>
</tbody>
</table>