

Subject: MCR10089 - Revise help, help_ and add verify_info, info_seg_
Author: Gary Dixon
Date: 2021-03-01
Multics Ticket: [#220](#)

[MTB-1004](#) describes problems with validate_info_seg (vis), and discusses a possible replacement: verify_info (vi).

[MTB-1005](#) describes changes needed in the help_ subsystem, and proposes a major revision for this subsystem and its supporting routines.

This MCR requests an enhancement to implement these proposals by:

- updating the help command;
- replacing the help_ subsystem;
- adding a new info_seg_ subroutine to parse info segments; and
- replacing the validate_info_seg (vis) command with a new verify_info (vi) command and info_seg_verify_ subroutine.

Proposed Changes

Aside from the source files mentioned in MTB-1004 and MTB-1005, three source files for programs using the subsystem utilities (ssu_) were modified; and a new include file was added.

- Add declarations for ssu_\$execute_string and ssu_\$get_info_ptr support routines to:
 - ssu_request_dcls_.incl.pl1
 - ssu_subroutine_dcls_.incl.pl1
 - ssu_standalone_command_.incl.pl1
- Create a new include file for subroutines that may create their own standalone invocation of ssu_ but have no arguments or control arguments to be processed. It contains only declarations for ssu_ subroutine entry points plus the ssu_-focused standalone_cleanup_handler routine (identical to the routine in ssu_standalone_command_.incl.pl1).

The following components are included in this change.

- Bound Segment: bound_info_rtns_
 - Replace bound_info_rtns_.bind with a version that accommodates the changes below.
 - Replace help.pl1 as described in MTB-1005.
 - Delete the old support routine help_rql_.pl1.
 - Replace the help_ subsystem as described in MTB-1005:
 - help_.pl1 (a highly-revised replacement)

- Add new help_subsystem support routines as described in MTB-1005:
 - help_listen_util.pl1
 - help_request_tables.alm
 - help_requests.pl1
 - help_responses.pl1
 - help_util.pl1
- Add new info_seg_subroutine and its support modules as described in MTB-1004:
 - info_seg.pl1
 - info_seg_allocate.pl1
 - info_seg_error.pl1
 - info_seg_parse.pl1
 - info_seg_specifications.cds
 - info_seg_util.pl1
- Add new verify_info command and supporting routines as described in MTB-1004:
 - verify_info.pl1
 - info_seg_verify.pl1
 - info_seg_verify_iFile.pl1
- Include files supporting the above changes:
 - Add _help_shared_data.incl.pl1 as described in MTB-1005.
 - Replace help_args.incl.pl1 as described in MTB-1005.
 - Replace help_cis_args.incl.pl1 as described in MTB-1005.
 - Add info_seg_dcls.incl.pl1 as described in MTB-1004.
 - Add verify_info_data.incl.pl1 as described in MTB-1004.
- Info segments supporting the above changes:
 - Delete several info segments which are being incorporated as info blocks in help.info:
 - help_infos.gi.info
 - help_responses.info
 - help_system.gi.info
 - topic.info
 - modules.gi.info
 - documentation.gi.info
 - Replace help.info as described in MTB-1005. It includes info blocks for the deleted info segs listed immediately above.
 - Replace help_.info as described in MTB-1005.
 - Replace list_help.info to modernize syntax style and better describe the TOPIC argument.
 - Add info segments for verify_info and the info_seg_subroutine as described in MTB-1004.
 - info_seg.info
 - info_seg_verify.info
 - verify_info.info

The following build script describes the specifics for each change noted above.

Installation_directory: >udd>m>gd>w>vi_07;

Build_script: vi_07.mb;

Bound_obj:	bound_info_rtns_	IN: sss	UPDATE;
bindfile:	bound_info_rtns_.bind		REPLACE;
source:	help.pl1	REPLACE	compiler: pl1 -ot;
source:	help_.pl1	REPLACE	compiler: pl1 -ot;
source:	help_rql_.pl1	DELETE;	
source:	help_listen_util_.pl1	ADD	compiler: pl1 -ot;
source:	help_request_tables_.alm	ADD	compiler: alm;
source:	help_requests_.pl1	ADD	compiler: pl1 -ot;
source:	help_responses_.pl1	ADD	compiler: pl1 -ot;
source:	help_util_.pl1	ADD	compiler: pl1 -ot;
source:	info_seg_.pl1	ADD	compiler: pl1 -ot;
source:	info_seg_allocate_.pl1	ADD	compiler: pl1 -ot;
source:	info_seg_error_.pl1	ADD	compiler: pl1 -ot;
source:	info_seg_parse_.pl1	ADD	compiler: pl1 -ot;
source:	info_seg_specifications_.cds	ADD	compiler: cds;
source:	info_seg_util_.pl1	ADD	compiler: pl1 -ot;
source:	info_seg_verify_.pl1	ADD	compiler: pl1 -ot;
source:	info_seg_verify_iFile_.pl1	ADD	compiler: pl1 -ot;
source:	verify_info.pl1	ADD	compiler: pl1 -ot;
Include:	_help_shared_data_.incl.pl1	IN: sss.include	ADD;
Include:	help_args_.incl.pl1	IN: sss.include	REPLACE;
Include:	help_cis_args_.incl.pl1	IN: sss.include	REPLACE;
Include:	info_seg_dcls_.incl.pl1	IN: sss.include	ADD;
Include:	ssu_invoker_dcls_.incl.pl1	IN: sss.include	ADD;
Include:	ssu_request_dcls_.incl.pl1	IN: sss.include	REPLACE;
Include:	ssu_standalone_command_.incl.pl1	IN: sss.include	REPLACE;
Include:	ssu_subroutine_dcls_.incl.pl1	IN: sss.include	REPLACE;
Include:	verify_info_data.incl.pl1	IN: sss.include	ADD;
Info:	help_infos.gi.info	IN: sss.info	DELETE;
Info:	help_responses.info	IN: sss.info	DELETE;
Info:	help_system.gi.info	IN: sss.info	DELETE;
Info:	topic.info	IN: sss.info	DELETE;
Info:	modules.gi.info	IN: sss.info	DELETE;
Info:	documentation.gi.info	IN: sss.info	DELETE;

```

Info:          help.info                               IN: sss.info  REPLACE;
              add_name:  help_infos.gi.info
                        help_infos.info
                        help_system.gi.info
                        help_system.info
                        help_responses.gi.info
                        help_responses.info
                        hr.gi.info
                        hr.info
                        topics.gi.info
                        topics.info
                        topic.gi.info
                        topic.info
                        modules.gi.info
                        modules.info
                        module.gi.info
                        module.info
                        documentation.gi.info
                        documentation.info
                        doc.gi.info
                        doc.info;

Info:          help_.info                               IN: sss.info  REPLACE;
Info:          list_help.info                           IN: sss.info  REPLACE;
              add_name:  lh.info;

Info:          info_seg_.info                           IN: sss.info  ADD;
Info:          info_seg_verify_.info                    IN: sss.info  ADD;
Info:          verify_info.info                          IN: sss.info  ADD;
              add_name:  vi.info;

```

Testing

The code described above was tested according to: the *Testing verify_info* section of MTB-1004; or the *Test Strategies* section of MTB-1005.

Results of testing for verify_info are described in MTB-1004.

Results of *Command Interface Testing* for the revised help command were completed successfully. The only difference between operation of the existing help command versus its slightly-improved replace were due to changes to handling of help control arguments, as documented in MTB-1005.

Results of *Subsystem Interface Testing* with the help_ subroutine interface was tested in three ways: called by the help command; called by an ssu_ subsystem using the ssu_\$help_request (the help request in most ssu_ subsystems); called by a subsystem which invokes help_ entry points directly.

- For ssu_ subsystems, the forum and analyze_multics (azm) subsystems were used for testing.
- For subsystems directly calling help_ entry points, mbuild and probe were used for testing.

All tests showed interfaces with the `help_` subroutine working as expected (almost identically whether the existing or revised `help_` subroutine was used). The only differences in behavior stemmed from bugs fixed only in the revised code, or new/changed control flags causing differences by design (fixing of bug, or implementation of new/changed feature behavior). Feature selection and searches for info blocks worked identically for the existing `help_` subroutine and the revised version of that subroutine.

Results of *Subroutine Testing* focused on whether the expected paragraphs were displayed as `help_` walked straight through an info segment. Were prompts issued before each terminal page filled up? Were any paragraphs missing or repeated? Was contents of any paragraph garbled? To some extent, this testing compared output from the existing `help_` subroutine with that from the revised subroutine; those tests show the same paragraphs being displayed without garble by both help versions.

However, the current help follows a different algorithm for determining how many paragraphs to display before the next prompt is made. Comparison of expected paragraphs output showed the expected differences between the two `help_` versions, with the revised `help_` displaying more paragraphs/sections on each terminal page and therefore issuing few prompts to the user.

Results of *Response Testing* focused on whether each user response to “More help?” prompt caused the documented function to be carried out. While many of the response routines for current and revised help are named the same, all of the response routines in revised `help_` were re-implemented as `ssu_` requests. And many of those `ssu_` response routines have additional control arguments not present in the existing `help_` subroutine. As a result, all controls for each revised `help_` request were tested for correct operation according to documentation for that response (displayable via the `help_` “help” response). That testing was successfully conducted as each request was developed.

The final audit of response code will ensure the code implementation agrees with documentation for that request appearing in:

- `help_request_tables_.alm` (source for `list_responses` description of responses)
- `? (summarize_responses)` output
- Comment at the start of `help_responses_.pl1` (patterned like output of `? request`)
- Comment at the beginning of code for each response entry point.
- Response documentation in `help_responses.gi.info`.
- Response documentation in per-response info blocks in `help.info`.
- MTB-1005 documentation for each response.

Documentation

Pertinent documentation appears in MTB-1004 for: the `verify_info` command; and for the `info_seg_` and `info_seg_verify_` subroutines.

Pertinent documentation changes appear in MTB-1005 for the `help` command; and for the `help_` subroutine. While new interfaces were added to the `info_seg_` subroutine for use by `help_`, those are internal interfaces which are not retained in the `bound_info_rtns_` bound segment. These interfaces are only documented as comments in the `info_seg_` code.

Version History

Date	Revision	Author	Comment
2021-02-22	1.0	Gary Dixon	First draft of changes, for preliminary review.
2021-02-23	1.1	Gary Dixon	Final draft for review by Multics community.
2021-03-01	1.2	Gary Dixon	Fix typo in Subject line.