Subject: MCR10139, Modernize Content in Search-Path-related Info Segs

Author: Gary Dixon

Date: 2023-09-10

Ticket: https://multics-trac.swenson.org/ticket/315

The add_search_paths.info and set_search_paths.info segments are missing certain details about the -initiated_segments and -referencing_dir keywords, and have unclear specifications for syntax and semantics of the command. These problems should be corrected. And the related delete_search_paths.info, print_search_paths.info and where_search_paths.info segment contents should be modernized to match changes in asp.info and ssp.info.

Proposed Changes

```
Description --
 1) Add/clarify missing/unclear info in add_search_paths.info and
    set_search_paths.info.
 2) Revise all five search_path-related command/AFs according to modern
    info seg formatting rules.


Changes --


 Info:         add_search_paths.info            IN: sss.info  REPLACE;
                add_name:  asp.info;
 Info:         delete_search_paths.info         IN: sss.info  REPLACE;
                add_name:  dsp.info;
 Info:         print_search_paths.info          IN: sss.info  REPLACE;
                add_name:  psp.info;
 Info:         set_search_paths.info            IN: sss.info  REPLACE;
                add_name:  ssp.info;
 Info:         where_search_paths.info          IN: sss.info  REPLACE;
                add_name:  wsp.info;
```

Testing

No code is changing, so only info segment format validation was required. verify_info was run against all changed info segments. All met modern info segment requirements.

Documentation

Proposed content for all five info segments is shown here.

```
>udd>m>gd>w>spi>add_search_paths.info   (108 lines in info)

2023-09-09  add_search_paths, asp

Syntax as a command:
  asp SEARCH_LIST SEARCH_PATH1 {-position_control_arg} ...
       SEARCH_PATHn {-position_control_arg} {-control_args}

Function:  adds one or more search paths to the specified search list.

Arguments:
SEARCH_LIST
   is the name of the search list to which the new search paths are
   added.
SEARCH_PATH
   specifies a new search path, where SEARCH_PATH is a relative or
   absolute pathname of a directory to be search, or a keyword
   designating such directory.  See "List of keywords" and
   "Notes on search paths" below for more details.

Control arguments (positioning a search path):
   One of the following may be given after each new SEARCH_PATH
   argument to specify a position within the existing search list for
   the new SEARCH_PATH.  The STR operand is a pathname or keyword as
   given in the existing search list.
-first, -ft
   positions the new search path first in the search list.
-last, -lt
   positions the new search path last in the search list.  (Default)
-after STR, -af STR
   positions the new search path after the search path denoted by STR.
-before STR, -be STR
   positions the new search path before the search path denoted by STR.

Control arguments:
-force, -fc
   causes duplicate pathnames to be replaced in the search list at the
   specified position.
-no_force, -nfc
   causes duplicate pathnames to be ignored.  (Default)
-inhibit_error, -ihe
   suppresses warning messages printed when a pathname is nonexistent
   or is already in the search list.
-no_inhibit_error, -nihe
   causes warning messages to be printed.  (Default)

List of keywords:
   The following keywords are accepted as search paths
   in place of absolute or relative pathnames.
-home_dir, -hd
   The user's home directory.
```

```
-process_dir, -pd
   The user's process directory.
-working_dir, -wd
   The users current working directory.

-initiated_segments, -is
   Represents the list of segments already known to the process by
   their reference name(s).  This keyword may only be used for the
   linker search list.
-referencing_dir, -rd
   Represents a directory designated by the program using the
   SEARCH_LIST to locate segments.  See "Notes on -referencing_dir"
   below.

List of related commands:
   delete_search_paths, dsp
   print_search_path, psp
   set_search_paths, ssp
   where_search_paths, wsp

Notes on search paths:
A SEARCH_PATH can be specified as a quoted string which includes the
[user project] and/or [user name] active function.  Use of such quoted
string delays expansion of the active string until searching actually
begins.  This feature allows creation of search paths that select
segments in or below the project directory or home directory of the
current user.  If the quoted string identifies a directory that does
not exist, the missing directory is skipped; searching for a matching
segment continues with the next rule is the SEARCH_LIST.

For example, a project administrator could specify use of a
project_start_up.ec by users logged in on her project; for details,
type: help process_overseer.gi

This project_start_up.ec could setup various search lists that include
segments tailored for use by that project.  These search lists include
a quoted SEARCH_PATH with ">udd>[user project]>..." to specify a
directory containing project-related tools, info segments, or
exec_coms.  The [user project] active string would be evaluated only
when a search was actually made for a tool, info segment, or exec_com.

Notes on -referencing_dir:
Each subroutine that wants to locate segments using a search list may
tell search_paths_$get what directory pathname to use if the search
list includes a -referencing_dir search path.  If this directory
pathname is an empty string, the -referencing_dir rule is skipped.

For example, when the linker is called to dynamically link a program to
another subroutine, it uses pathname of the program containing the
unsnapped link as its -referencing_dir.  This allows a modified
program to to call a new (not yet installed) subroutine if the
programmer places that new subroutine (or a symbolic link to that
subroutine) in the directory containing the modified program.

To use the -referencing_dir rule in the linker search list:
   asp linker -referencing_dir -after -initiated_segments
```

```
To use the -referencing_dir rule in the info_segments search list:
   asp info_segments -referencing_dir -after -working_dir



>udd>m>gd>w>spi>delete_search_paths.info   (45 lines in info)

2023-09-09  delete_search_paths, dsp

Syntax as a command:  dsp SEARCH_LIST SEARCH_PATHs {-control_arg}

Function: allows you to delete one or more search paths from the
specified search list.

Arguments:
SEARCH_LIST
   is the name of the search list from which the specified search paths
   are deleted.
SEARCH_PATH
   specifies a search path to be deleted.  The search path must be an
   absolute or relative pathname or a keyword (see "List of keywords"
   below).  Quote the pathname if it contains spaces, active strings,
   or other command language characters.

Control arguments:
-all, -a
   specifies that the search list itself is to be deleted.  Any search
   paths specified are ignored.  Use -all to delete all the search
   paths in a search list.

List of keywords:
   The following keywords are accepted as search paths
   in place of absolute or relative pathnames.
-home_dir, -hd
   The user's home directory.
-process_dir, -pd
   The user's process directory.
-working_dir, -wd
   The users current working directory.

-initiated_segments, -is
   Represents the list of segments already known to the process by
   their reference name(s).  This keyword may only be used for the
   linker search list.
-referencing_dir, -rd
   Represents a directory designated by the program using the
   SEARCH_LIST to locate segments.  See "Notes on -referencing_dir"
   below.

Notes:
For a complete list of the search facility commands, type:
  help add_search_paths -scn command
```

```
>udd>m>gd>w>spi>print_search_paths.info   (22 lines in info)

2023-09-09  print_search_paths, psp

Syntax as a command:  psp {SEARCH_LISTs} {-control_args}

Syntax as an active function:  [psp SEARCH_LIST {-control_args}]

Function: prints the search paths in specified search lists.  As an
active function, returns the ordered list of pathnames and/or keywords
in the given search list.

Arguments:
SEARCH_LIST
   name of a search list to print.  If none are specified, all search
   lists referenced in the process are printed.

Control arguments:
-expanded, -exp
   print all keywords except -referencing_dir and all unexpanded search
   paths as absolute pathnames.

Notes: All synonyms of a search list name are printed if no search
lists are specified.



>udd>m>gd>w>spi>set_search_paths.info   (93 lines in info)

2023-09-09  set_search_paths, ssp

Syntax:  ssp SEARCH_LIST {SEARCH_PATHs} {-control_args}

Function:  allows a user to replace the search paths contained in a
specified search list.  If no SEARCH_PATHs are specified, then the
specified SEARCH_LIST is set to its system-defined default list of
search paths.  It is an error to create an empty SEARCH_LIST for which
no defaults are defined by the system.

Arguments:
SEARCH_LIST
   is the name of a search list.  If this search list does not exist,
   it is created.  A warning message is printed if a search list is
   created and it is not system-defined.
SEARCH_PATHs
   are one or more search paths to be placed in the specified
   SEARCH_LIST.  Each SEARCH_PATH appears in the order specified in
   this command.  SEARCH_PATH may be an absolute or relative pathname
   of a directory to be searched, or a keyword designating such
   directory.  See "List of keywords" and "Notes on search paths" below
   for more details.

Control arguments:
-brief, -bf
   suppresses a warning message for the creation of a search list not
   defined by the system.
```

```
-default, -df
   replaces the SEARCH_LIST with its system-defined default search
   paths.  No SEARCH_PATH arguments may be specified with this control
   argument.

List of keywords:
   The following keywords are accepted as search paths
   in place of absolute or relative pathnames.
-home_dir, -hd
   The user's home directory.
-process_dir, -pd
   The user's process directory.
-working_dir, -wd
   The user's working directory when searching begin.

-initiated_segments, -is
   Represents the list of segments already known to the process by
   their reference name(s).  This keyword may only be used for the
   linker search list.
-referencing_dir, -rd
   Represents a directory designated by the program using the
   SEARCH_LIST to locate segments.  See "Notes on -referencing_dir"
   below.

Notes on search paths:
A SEARCH_PATH can be specified as a quoted string which includes the
[user project] and/or [user name] active function.  Use of such quoted
string delays expansion of the active string until searching actually
begins.  This feature allows creation of search paths that select
segments in or below the project directory or home directory of the
current user.  If the quoted string identifies a directory that does
not exist, the missing directory is skipped; searching for a matching
segment continues with the next rule is the SEARCH_LIST.

For example, a project administrator could specify use of a
project_start_up.ec by users logged in on his project; for details,
type: help process_overseer.gi

This project_start_up.ec could setup various search lists that include
segments tailored for use by that project.  These search lists include
a quoted SEARCH_PATH with ">udd>[user project]>..." to specify a
directory containing project-related tools, info segments, or
exec_coms.  The [user project] active string would be evaluated only
when a search was actually made for a tool, info segment, or exec_com.

Notes on -referencing_dir:
Each subroutine that wants to locate segments using a search list may
tell search_paths_$get what directory pathname to use if the search
list includes a -referencing_dir search path.  If this directory
pathname is an empty string, the -referencing_dir rule is skipped.

For example, when the linker is called to dynamically link a program to
another subroutine, it uses pathname of the program containing the
unsnapped link as its -referencing_dir.  This allows a modified
program to to call a new (not yet installed) subroutine if the
programmer places that new subroutine (or a symbolic link to that
subroutine) in the directory containing the modified program.
```

```
To use the -referencing_dir rule in the linker search list:
   asp linker -referencing_dir -after -initiated_segments

To use the -referencing_dir rule in the info_segments search list:
   asp info_segments -referencing_dir -after -working_dir

Notes:  For a complete list of the search facility commands, type:
  help add_search_paths -scn command




>udd>m>gd>w>spi>where_search_paths.info   (33 lines in info)

2023-09-09  where_search_paths, wsp

Syntax as a command:  wsp SEARCH_LIST ENTRYNAME {-control_arg}

Syntax as an active function:
   [wsp SEARCH_LIST ENTRYNAME {-control_args}]

Function: prints or returns the absolute pathname of ENTRYNAME
found using the current search rules in the given SEARCH_LIST.

Arguments:
SEARCH_LIST
   is the name of the search list searched.
ENTRYNAME
   is the entryname sought.  Each name must include the segment suffix
   appropriate for items in the search list.

Control arguments:
-all, -a
   prints or returns all occurrences of this ENTRYNAME found by probing
   every directory in the SEARCH_LIST.
-inhibit_error, -ihe
   do not print an error if no occurrences of the entryname are found.
-no_inhibit_error, -ihe
   print an error if no occurrences of the entryname are found.
   (Default)

Access required: You must have s access on the directory being
searched, or nonnull access to the segment matching ENTRYNAME.

Notes:
For a complete list of the search facility commands, type:
  help add_search_paths -scn command
```

Version History

Date	Revision	Author	Comment
2023-09-09	1.0	Gary Dixon	Initial draft of this MCR.
2023-09-10	1.1	Gary Dixon	Include set_search_paths.info missing from v1.0 of this MCR.