

Subject: MCR-10082, get_page_length_
Author: Gary Dixon
Date: 2020-07-06

There should be a `get_page_length_` function that functions similar to the existing `get_line_length_` function. For a given I/O switch, it would return the current length from the plNN mode string, or would return a default page length of 23 for switches not supporting the plNN mode. This suggestion is reference in Multics change ticket:

<http://multics-trac.swenson.org/ticket/211>

This MCR proposes installing this new function as a new component in `bound_io_commands_`.

Proposed Changes

Add a new `get_page_length_.pl1` function which returns the page length value from the I/O mode string for a given I/O switch.

```
help get_page_length_$(stream switch) -brief
```

Syntax:

```
declare get_page_length_$(stream entry (char(*), fixed bin(35)) returns  
    (fixed bin);  
page_length = get_page_length_$(stream (switch_name, code);
```

Syntax:

```
declare get_page_length_$(switch entry (ptr, fixed bin(35)) returns  
    (fixed bin);  
page_length = get_page_length_$(switch (switch_ptr, code);
```

The new procedure calls the existing `mode_string_` subroutine to find/extract the plNN value from the I/O stream's mode string. If the given I/O stream does not support the mode operation, or does not contain a plNN mode value, then a default page length of 23 is returned. 23 was chosen as a reasonable terminal page length for a minimum-sized terminal; it matches the default line length of 71 returned by `get_line_length_`.

`get_page_length_` would be installed in the same bound object as `get_line_length_`. It must be installed as a component separate from the existing `get_line_length_` because both procedures have entrypoints named the same that return different values.

pl.mb 07/06/20 0947.3 pdt Mon

Description:

A) Add get_page_length_ subroutine to bound_io_commands_.

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

Build_script: pl.mb;

Bound_obj:	bound_io_commands_	IN: sss UPDATE;
bindfile:	bound_io_commands_.bind	REPLACE;
source:	get_page_length_.pl1	ADD compiler: pl1 -ot;

Info:	get_page_length_.info	IN: sss.info ADD;
-------	-----------------------	-------------------

Testing

All entrypoints of this subroutine were tested on I/O switches attached via tty_ and window_io_ using the call command. The returned page length value compared correctly with the plNN mode string value.

Documentation

The info segment for this new function is shown below.

:Info: get_page_length_: 2020-07-06 get_page_length_

Function: returns the page length currently in effect on a given I/O switch. If the page length is not available (for any reason), a status code is returned, and a default page length is returned.

Entry points in get_page_length_:
(List is generated by the help command)

:Entry: stream: 2020-07-06 get_page_length_\$stream

Function: This entry point returns the page length of a given I/O switch, identified by name.

Syntax:
declare get_page_length_\$stream entry (char(*), fixed bin(35)) returns
 (fixed bin);
page_length = get_page_length_\$stream (switch_name, code);

Arguments:**switch_name**

is the name of the switch whose page length is desired. (Input) If switch_name is an empty string, the user_output I/O switch is assumed.

code

is a standard status code. (Output)

page_length

is the page length of switch_name. (Output)

:Entry: switch: 2020-07-06 get_page_length_\$switch

Function: This entry point returns the page length of a given I/O switch, identified by pointer.

Syntax:

declare get_page_length_\$switch entry (ptr, fixed bin(35)) returns
(fixed bin);

page_length = get_page_length_\$switch (switch_ptr, code);

Arguments:**switch_ptr**

is a pointer to the I/O control block of the switch whose page length is desired. (Input) If switch_ptr is null, the user_output I/O switch is assumed.

code

is a standard status code. (Output)

page_length

is the page length of switch_name. (Output)

Version History

Date	Revision	Author	Comment
2020-07-06	1.0	Gary Dixon	Initial version of MCR.