Subject: MCR-10036, cv_ptr_ virtual_ptr to Ring 0 Segment
Author: Gary Dixon
Date: February 23, 2017

Many tools could benefit by enhancing cv_ptr_ to accept a virtual_ptr argument to a hardcore (ring 0) segment. Currently, cv_ptr_ rejects a virtual_ptr argument if the constructed pointer is rejected by hcs$_fs_get_get_path_name. The new pointer_info tool uses cv_ptr_ to convert its input argument from a virtual_ptr to an actual pointer. The example below shows the problem encountered when giving a virtual_ptr referencing a ring 0 segment number. The new pointer_info tool tries to call cv_ptr_ to convert an input virtual_ptr to an actual pointer:

```
list_ref_names 75
  75 restart_fault (ring 0)
r 05:11 0.022 0

pointer_info 75|0
pointer_info: There was an attempt to use an invalid segment number.
  Converting ptr: 75|0
r 21:35 0.125 3
```

A similar problem occurs when attempting to use a reference name to construct a pointer, when that name is defined only within ring 0:

```
pointer_info restart_fault$1 -a
pointer_info: The name was not found. Converting ptr: restart_fault$1
r 05:06 0.045 4
```

- For details, reference Multics Ticket: [http://multics-trac.swenson.org/ticket/45](http://multics-trac.swenson.org/ticket/45)

### Proposed Changes

Change >ldd>sss>s>bound_conversion_rtns_.s::cv_ptr_.pl1:

- When converting a virtual_ptr beginning with a reference name, if hcs$_fs_get_seg_ptr returns error_table$_name_not_found, then try calling ring0_get$_segptr to evaluate that reference name.

- When validating a virtual_ptr beginning with a segment number, check the segment number against the first_stack segment number (2nd argument returned by hcs$_high_low_seg_count). If segment number < first_stack, then call ring0_get$_name to validate the segment number, rather than hcs$_fs_get_path_name.

- For hardcore segments, don’t attempt to call object_lib$_initiate or initiate_file$_component to look inside a bound segment or an archive. Such are not supported as ring 0 segments; and the user may not have access to look inside hardcore segments.
For example, the enhanced cv_ptr_ can now construct a pointer to a gate into ring 0, using its ring 0 segment number.

```
pointer_info 75|0
  For pointer: 75|0
    information:     restart_fault$0 (ring 0)
```

The same tool can reference the gate entrypoint by reference name, even though this name is defined only inside ring 0:

```
list_ref_names restart_fault
list_ref_names: Entry not found. >user_dir_dir>SysEng>G Dixon>restart_fault
```

```
pointer_info restart_fault$1 -a
  For pointer: 75|1
    information:     restart_fault$1 (ring 0)
```

```
  octal pointer:   000075400043 000001000000
  path: restart_fault
  segment: restart_fault
  offset: 1
  ring: 0
```

**Documentation**

No documentation is affected by this change. The cv_ptr_ subroutine writeup in the Multics Subroutines Manual, AG93-05A does not mention supporting or excluding ring 0 segments. So readers might expect that a virtual_ptr in any of its forms could identify a ring 0 segment. From that expectation, this change might be viewed as a defect repair, rather than an enhancement.

**Version History**

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Author</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-02-20</td>
<td>1.0</td>
<td>Gary Dixon</td>
<td>Initial draft of this MCR.</td>
</tr>
<tr>
<td>2017-02-23</td>
<td>1.1</td>
<td>Gary Dixon</td>
<td>Correct editing errors.</td>
</tr>
</tbody>
</table>