MCR10214
Add Support for Specifying Segment Number on initiate Command Line
Revision 1.0

Eric Swenson

November 23, 2026

1 Problem Description

Ticket #230 requests an enhancement to the `initiate` command to allow the specification of a desired segment number. This is currently not supported, although the `hcs_$initiate` subroutine can do this. The subroutine allows specifying the segment switch `seg_sw` on and the input `seg_ptr` gives the segment number.

2 Proposed Fix

New control argument `-segment_number N` (`-segno N`), N in octal (as `initiate -long` prints segment numbers), from 0 to 77777.

- The value is checked with `cv_oct_check_`; a missing or bad value is reported (`error_table_$noarg`, `$bad_conversion`). The second pass over the arguments, which takes the path and reference names, skips the value.
- Each `hcs_$initiate` call passes `seg_sw = 1` and `seg_ptr = baseptr (N)`. The hardcore (`makeknown_`) rejects a number that is in use (`error_table_$segno_in_use`) or not valid for the ring (`$invalidsegno`); these are reported like any other initiation error.
- If the segment is already known with another segment number, `hcs_$initiate` returns `error_table_$segknown` and the existing number. `initiate` then says so once ("PATH is already known with segment number X, not N.") and goes on initiating the reference names.
- Further reference names (`-all`, or several names) find the segment already known with that number, as usual.

The `compare_ascii` comparison of the current version of `initiate.pl1` and the proposed version is below:

```
Inserted in B:
B50       dcl (segno_sw, segno_warned) bit (1);
B51       dcl req_segno fixed bin (35);
```

```
B52      dcl MAX_SEGNO fixed bin (35) int static options (constant) init (32767); /*
    15-bit segment number */
Preceding:
A50
A51      dcl (area_ptr, arg_ptr, names_ptr, seg_ptr) ptr;


Inserted in B:
B59      dcl error_table_$bad_conversion fixed bin (35) ext;
Preceding:
A56      dcl error_table_$badopt fixed bin (35) ext;


Inserted in B:
B63      dcl error_table_$noarg fixed bin (35) ext;
Preceding:
A59      dcl error_table_$noalloc fixed bin (35) ext;


Inserted in B:
B72      dcl cv_oct_check_ entry (char (*), fixed bin (35)) returns (fixed bin (35));
Preceding:
A67      dcl expand_pathname_ entry (char (*), char (*), char (*), fixed bin (35));


A75      dcl (addr, baseno, bin, fixed, null, ptr, rtrim, substr) builtin;
Changed by B to:
B81      dcl (addr, baseno, baseptr, bin, fixed, null, ptr, rtrim, substr) builtin;


A85               all_sw, force_sw, long_sw, some_args = "0"b;
Changed by B to:
B91               all_sw, force_sw, long_sw, some_args, segno_sw, segno_warned = "0"b;


Inserted in B:
B106                  else if arg = "-segment_number" | arg = "-segno" then do;
B107                       if i = arg_count then do;
B108                            call com_err_ (error_table_$noarg, "initiate", "Following
    ^a.", arg);
B109                            return;
B110                       end;
B111                       i = i + 1;
B112                       call cu_$arg_ptr (i, arg_ptr, arg_len, code);
B113                       req_segno = cv_oct_check_ (arg, code);
B114                       if code ^= 0 | req_segno < 0 | req_segno > MAX_SEGNO then do;

Inserted in B:
B106                  else if arg = "-segment_number" | arg = "-segno" then do;
B107                       if i = arg_count then do;
B108                            call com_err_ (error_table_$noarg, "initiate", "Following
    ^a.", arg);
```

```
B109                            return;
B110                        end;
B111                        i = i + 1;
B112                        call cu_$arg_ptr (i, arg_ptr, arg_len, code);
B113                        req_segno = cv_oct_check_ (arg, code);
B114                        if code ^= 0 | req_segno < 0 | req_segno > MAX_SEGNO then do;
B115                            call com_err_ (error_table_$bad_conversion, "initiate",
B116                                "Segment number must be octal, from 0 to ^o: ^a",
    MAX_SEGNO, arg);
B117                            return;
B118                        end;
B119                        segno_sw = "1"b;
B120                    end;
Preceding:
A100                    else do;


Inserted in B:
B139                    if arg = "-segment_number" | arg = "-segno" then i = i + 1; /* skip
    its value */
Preceding:
A118                    if substr (arg, 1, 1) = "-" then go to NEXT_ARG;


A193     INITIATE: call hcs_$initiate (act_dn, act_en, refname, 0, 0, seg_ptr, code);
Changed by B to:
B215     INITIATE: if segno_sw then seg_ptr = baseptr (req_segno); /* input: segment
    number to use */
B216                call hcs_$initiate (act_dn, act_en, refname, fixed (segno_sw, 1), 0,
    seg_ptr, code);


Inserted in B:
B244                else if segno_sw & ^segno_warned & bin (baseno (seg_ptr), 17) ^=
    req_segno then do;
B245                                                            /* already known, with
    another number */
B246                    call com_err_ (0, "initiate", "^a^[>^]^a is already known with
    segment number ^o, not ^o.",
B247                        dn, dn ^= ">", en, bin (baseno (seg_ptr), 17), req_segno);
B248                    segno_warned = "1"b;
B249                end;
B250
Preceding:
A221                else if long_sw then do;                    /* success */

Comparison finished: 10 differences, 36 lines.
```

3 Documentation

`initiate.info`: new entry for `-segment_number N, -segno N`; date updated.

The `compare_ascii` of the current `initiate.info` and the proposed version is below:

```
cpa >doc>info>initiate.info ==

A1       07/25/86 initiate, in
Changed by B to:
B1       :Info: initiate: in: 2026-11-23 initiate, in
B2


Inserted in B:
B39      -segment_number N, -segno N
B40         makes the file known with segment number N (octal), which must not
B41         be in use. If the file is already known with another segment
B42         number, a message gives that number and the file is initiated by
B43         the ref_names under it.
Preceding:
A38
A39
A40      Access required: nonnull.


A43      Notes: Initiating a multisegment file involves initiating component 0 of
A44      the MSF with the reference names specified.
Changed by B to:
B49      Notes: Initiating a multisegment file involves initiating component 0
B50      of the MSF with the reference names specified.

Comparison finished: 3 differences, 12 lines.
```

4 Test

1. `initiate X -segno 1000 -lg`: X initiated with segment number 1000; `where X` / `list_ref_names` agree.
2. The same with a number already in use: `segno_in_use` error. With a number below the lowest user segment number: `invalidsegno`.
3. `initiate X -segno 1001` when X is already known: the "already known" message.
4. `-segno` with no value, with `8`, with `100000`: usage errors. `initiate X Y -segno 1002 -all`: all names.

5 Files Changed

bound_ti_term_.s.archive	initiate.pl1
>sss	bound_ti_term_ (rebound)
>doc>info	initiate.info

6 Version History

2026-11-23	1.0	Initial version of the MCR.