MCR10208
Fix Uninitialized FSB Pointer for PL/I get string
Revision 1.0

Eric Swenson

November 22, 2026

1 Problem Description

Ticket #206 describes an issue where a PL/I statement such as

```
get string (translate (dottedaddr, " ", ".")) list (fields);
```

usually works but sometimes takes a command fault in the PL/I I/O runtime (`bound_plio2_`). Charles Anthony traced it, by filling the stack with fixed values, to an `lprp` instruction loading an uninitialized packed pointer: if the two high bits of the word happen to be set, the instruction faults.

1.1 Cause

For the STRING option there is no file. The compiled code passes a "fake" file state block (FSB), short and on the stack, whose buffer pointer it sets to the string; `plio2_get_util_` (and `plio2_put_util_` for `put string`) fills in the rest of the fields it needs, but not `fsb.iocb_p`, the packed pointer (`ptr unaligned`, word 3) to the file's I/O switch. Other code copies that field without using it (`plio2_gvl_`: `iocbp = ps.fsbp -> fsb.iocb_p;`), and the copy loads it with `lprp`, which faults on some bit patterns. Since the value is never used for a string, the results are otherwise correct, which is why the failure is intermittent.

2 Proposed Fix

As found in the ticket, `fsb.iocb_p` is set to null when the fake FSB is set up for `get string` in `plio2_get_util_`, and likewise for `put string` in `plio2_put_util_` (whose buffer-flush path also loads `fsb.iocb_p`):

```
fsb.iocb_p=null;      /* no file: the fake FSB's iocb_p was never set, and
    loading
                  an arbitrary value with lprp can fault (ticket 206) */
```

`iocb_p` is word 3 of the FSB, before `bptr` (word 6), which the compiled code already sets in the fake FSB, so it lies within the fake FSB and is safe to store into.

The `compare_ascii` output below shows the changes for `plio2_put_util_.pl1`:

```
cpa [lpn plio2_put_util_.pl1] ==

Inserted in B:
B153                        fsb.iocb_p=null;  /* no file: the fake FSB's iocb_p was never
    set, and loading
B154                                             an arbitrary value with lprp can fault
    (ticket 206) */
Preceding:
A153
A154                        bsize=ps.number;

Comparison finished: 1 difference, 2 lines.
```

And the `compare_ascii` output below shows the changesz for `plio2_get_util_.pl1`:

```
cpa [lpn plio2_get_util_.pl1] ==

Inserted in B:
B494                        fsb.iocb_p=null;  /* no file: the fake FSB's iocb_p was never
    set, and loading
B495                                             an arbitrary value with lprp can fault
    (ticket 206) */
Preceding:
A494
A495                        if ps.varying_string then i=addrel(bptr,-1)->based_int;

Comparison finished: 1 difference, 2 lines.
```

3 Documentation

No documentation changes.

4 Test

1. The ticket's method: a program that fills the stack with words whose two high bits are set (for example by calling a procedure with a large automatic array of `"777777777777"b3` and returning), then executes the `get string` statement. With the old library it faults; with the new one it gives the correct result.
2. The same for `put string (s) list (...)`.
3. Ordinary `get`/`put` on files and the terminal (`sysin`/`sysprint`) are unchanged.

5 Files Changed

bound_plio2_.s.archive	plio2_get_util_.pl1, plio2_put_util_.pl1
>sss	bound_plio2_ (rebound)

6 Version History

2026-10-10	1.0	Initial version of the MCR.