

MCR10156

Prevent absurdly large values of **maxu** and fixed point overflow in **as_who** Revision 1.0

Eric Swenson

July 27, 2025

1 Problem Description

Multics ticket <https://multics-trac.swenson.org/ticket/343> describes an issue where setting the value of the load control parameter **maxu** using the operator command of the same name results in a fixed point overflow in the **as_who** command.

The storage for the **maxu** value is **fixed bin(17)**, and thus any value greater than 131071 for this parameter will be allowed by the compiler, but will result in a fixed point overflow when this value is used in floating point calculations, such as those that occur in the **as_who** command.

This MCR proposes to address this problem by having the **admin_** subroutine check the value provided by the operator **maxu** command to make sure it is not greater than 131071. If a larger value is provided, an appropriate error message is issued and no action is taken.

The fragment below shows operator interaction, below, shows the current behavior:

```
Ready
M-> maxu 99999999

admin_: maxu: 99999999 (99999999 normal users)
Ready
M-> who -lg

Error: Fixed point overflow by as_who$long|4006
(>system_library_tools>bound_admin_tools_|4006)
(actually by support procedure pl1_signal_$pl1_signal_from_ops_|776
(>sl1>bound_library_1_|13712)) (fixedoverflow condition)
who: Execution aborted by error signal.
Ready
```

2 Proposed Fix

The proposed fix is to add code to the `admin_` subroutine, which implements the operator `maxu` command to check for supplied values greater than 131071, and to issue an error message and take no action, rather than setting an invalid value.

The fix is trivial and can be seen in the `compare_ascii` output, below:

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

Inserted in B:

```
B667          if i > 131071 then do;                      /* check for value too big
               for fixed bin (17) */
B668                      answer = arg || " is too large a value for maxunits";
B669                      go to admin_answers;
B670          end;
Preceding:
A667          anstbl.max_units = i;                      /* set the system max load
               units. */
```

Inserted in B:

```
B3477      admin_: maxunits: error: XXX is too large a value for maxunits
B3478
B3479      S:      $as1
B3480
B3481      T:      In response to an operator maxunits command.
B3482
B3483      M:      The argument of the maxunits command cannot be larger than
B3484      131071. No action was taken.
B3485
B3486      A:      $tryagain
B3487
B3488      Message:
Preceding:
A3473      admin_: shift: error: XXXX is not a legal shift number.
```

Comparison finished: 2 differences, 16 lines.

The info segment for the operator `maxu` command, `maxunits.info` will also be updated in order to mention the upper limit:

```
cpa [wsp info maxunits.info] ==
```

```
A1          02/13/85 maxunits, maxu
Changed by B to:
B1          1985-02-13 maxunits, maxu
```

```
A15          than zero. If no argument is given, the current value is printed.
Changed by B to:
B15          than zero and less or equal to 131071. If no argument is given,
```

B16 the current value is printed.

Comparison finished: 2 differences, 5 lines.

3 Documentation

The error message documentation is updated, as can be seen in the above `compare_ascii` output. The updated info segment provides the updated online documentation of the `maxu` command.

4 Test

The changes will be tested by issuing the `maxu` command with various good and bad values, and examining the error message and effects.

5 Version History

2025-07-27	1.0	Initial version of the MCR.
------------	-----	-----------------------------