

MCR10015

Fix Date in Emacs `make-wall-chart` Header

Eric Swenson

December 25, 2016

1 Issue

Currently, the date displayed in the header of the Emacs buffer created by `make-wall-chart` has the format `12/26/116` due to the Lisp `date` function, defined in `e_macops.lisp`, not correctly handling dates in the 21st century. This MCR calls for a simple fix to cause dates in the 21st century to get formatted as `12/26/16`. This issue is described in the ticket:

<https://sourceforge.net/p/dps8m/tickets/107/>.

2 Proposed Change

The current definition of the `date` function is as follows:

```
(defun date () ;general utility BSG 10/31/79
  (let ((statdate (mapcar 'decimal-rep (status date))))
    (concatenate (cadr statdate)
                  "/"
                  (caddr statdate)
                  "/"
                  (car statdate))))
```

`(status date)` returns a list such as `(164 14 31)`, which represents the octal values for the year, month, and date. The expression `(mapcar 'decimal-rep (status date))` results in the list of three strings of the decimal representation of these values, thus `(/116 /12 /25)`. As can be seen, the current year, 2016, is represented as the string "116". The three values are concatenated, separated with slashes, to form a date string.

The proposed simple solution replaces the expression `(car statdate)` with this expression:

```
(maknam
```

```
(exploden (remainder (read-from-string (car statdate)) 100))).
```

The updated definition of the `date` function would therefore be:

```
(defun date () ;general utility BSG 10/31/79
  (let ((statdate (mapcar 'decimal-rep (status date))))
    (catenate
      (cadr statdate)
      "/"
      (caddr statdate)
      "/"
      (maknam
        (exploden
          (remainder
            (read-from-string (car statdate)) 100))))))
```

For those readers who don't know Multics MacLisp, working from inside to outside, the `read-from-string` turns the string into an integer, the `remainder` computes the remainder, modulo 100, thus converting 116 to 16, the `exploden` creates a list of the character representation of the number 16 (61, 66), and finally, the `maknam` creates a string from the list of characters. It is possible that Multics MacLisp provides a more elegant way of doing this, but the MacLisp documentation on string manipulation is very shallow and I could find no documentation on the required number-to-string and string-to-number functions to make this more straightforward.

The above change to the function `date` is made in the file `e_macops.lisp` in `bound_emacs_full.s.archive`. It is compiled (`lcp e_macops.lisp`) and `bound_emacs_full` is rebound. When changes to Lisp functions defined in `bound_emacs_full` are made, a new Emacs Lisp save image must be "dumped out". This is done using the `>unb>make_emacs.ec exec.com`. Both the `emacs` and `emacs_` Lisp saved images must be regenerated (see `>unb>make_emacs.ec` for details).

The updated artifacts are:

emacs.12.9.sv.lisp Full emacs saved image

emacs_.12.9.sv.lisp Minimal emacs saved image

bound_emacs_full_ Bound segment containing `e_macops_`

bound_emacs_full.s.archive Source archive for above

bount_emacs_full.archive Object archive for above