

**From:** Gareth Hunt <ghunt@cv3.cv.nrao.edu>  
**To:** abridle@cv3.cv.nrao.edu  
**Subject:** Fyi: Year in Modcomp OS  
**Date:** Fri, 18 Jul 1997 14:45:43 -0400 (EDT)

----- start of forwarded message (RFC 934 encapsulation) -----  
 From: George Martin <gmartin@aoc.nrao.edu>  
 To: ghunt@aoc.nrao.edu, rmilner@aoc.nrao.edu, gvanmoor@aoc.nrao.edu  
 Cc: ksowinsk@aoc.nrao.edu, bsahr@aoc.nrao.edu  
 Subject: Year in Modcomp OS  
 Date: Fri, 18 Jul 1997 11:14:59 -0600 (MDT)

During the meeting this morning when Y2K came up, you asked me to verify my memory about how the year number is store internally by the Modcomp OS. While the year is stored as a signed 16 bit integer, the algorithm which computes it is too simple. Note that the value for the year comes from the IAT clock.

```
*      GET THE YEAR NUMBER, FIRST TWO NIBBLES IN R4
*
LBR,R3,TWO          CONVERT TWO BCD NUMBERS
BLM,R8             GO CONVERT
ADI,R11            PRETTY IT UP
LAI,R1             GET BUFFER ADDR FOR STORAGE
STS,R11,YR         STORE THE YEAR FOR LATER
ETI,R11           TEST FOR LEAP YEAR
HZR,FEB.OK        IF EITHER OR BOTH BITS, NOT LEAP YEAR
ABMM,ONE          DIVISIBLE BY 4 SO AND DAY TO FEB
```

Given that the input is two BCD digits, there is no fool proof way of fixing this. But if we are still using the current system in 1999, this could be fixed by testing the value after conversion from BCD to something like 70, and if less then add 100. Then the code would work through the transition to 2000 and beyond. (Note the leap year test breaks in 2100!)

Looking at some code I "borrowed" from the OS later down to actually store the calendar date in the OS data structure, it looks like the system command that sets the date will work in the year 2000 and beyond if you enter a 4 digit year. If you enter a 2 digit year, it too will do the wrong thing.

I think that Ken has sugested that the only thing which will be affected are some time stamps on printers where the time is taken from a OS system call. All important observing data timestamps are derived from the MJAD. This makes the assumption that the MJAD is computed rigorously at system startup.

George  
 ----- end -----

**From:** Gareth Hunt <ghunt@cv3.cv.nrao.edu>  
**To:** abridle@cv3.cv.nrao.edu, rsimon@cv3.cv.nrao.edu  
**Subject:** Fyi: Year in Modcomp OS  
**Date:** Fri, 18 Jul 1997 15:59:01 -0400 (EDT)

Two messages re: VLA on-line system.

----- start of forwarded message (RFC 934 encapsulation) -----  
**From:** George Martin <gmartin@aoc.nrao.edu>  
**To:** ghunt@aoc.nrao.edu, rmilner@aoc.nrao.edu, gvanmoor@aoc.nrao.edu  
**Cc:** ksowinsk@aoc.nrao.edu, bsahr@aoc.nrao.edu  
**Subject:** Year in Modcomp OS  
**Date:** Fri, 18 Jul 1997 11:14:59 -0600 (MDT)

During the meeting this morning when Y2K came up, you asked me to verify my memory about how the year number is store internally by the Modcomp OS. While the year is stored as a signed 16 bit integer, the algorithm which computes it is too simple. Note that the value for the year comes from the IAT clock.

```
*      GET THE YEAR NUMBER, FIRST TWO NIBBLES IN R4
*
      LBR,R3,TWO          CONVERT TWO BCD NUMBERS
      BLM,R8              GO CONVERT
      ADI,R11             PRETTY IT UP
                        1900
      LAI,R1              GET BUFFER ADDR FOR STORAGE
                        P.BUF
      STS,R11,YR          STORE THE YEAR FOR LATER
      ETI,R11             TEST FOR LEAP YEAR
                        #FFFC
      HZR,FEB.OK          IF EITHER OR BOTH BITS, NOT LEAP YEAR
      ABMM,ONE           LEN.M+1      DIVISIBLE BY 4 SO AND DAY TO FEB
```

Given that the input is two BCD digits, there is no fool proof way of fixing this. But if we are still using the current system in 1999, this could be fixed by testing the value after conversion from BCD to something like 70, and if less then add 100. Then the code would work through the transition to 2000 and beyond. (Note the leap year test breaks in 2100!)

Looking at some code I "borrowed" from the OS later down to actually store the calendar date in the OS data structure, it looks like the system command that sets the date will work in the year 2000 and beyond if you enter a 4 digit year. If you enter a 2 digit year, it too will do the wrong thing.

I think that Ken has sugested that the only thing which will be affected are some time stamps on printers where the time is taken from a OS system call. All important observing data timestamps are derived from the MJAD. This makes the assumption that the MJAD is computed rigorously at system startup.

George  
 ----- end -----

----- start of forwarded message (RFC 934 encapsulation) -----  
**From:** Ken Sowinski <ksowinsk@aoc.nrao.edu>  
**To:** ghunt@aoc.nrao.edu, rmilner@aoc.nrao.edu, gvanmoor@aoc.nrao.edu,  
 gmartin@aoc.nrao.edu  
**Cc:** ksowinsk@aoc.nrao.edu, bsahr@aoc.nrao.edu  
**Subject:** Re: Year in Modcomp OS  
**Date:** Fri, 18 Jul 1997 13:15:04 -0600

George wondered:

> MJAD. This makes the assumption that the MJAD is computed rigorously at  
> system startup.

I just examined the relevant code and found that it takes the year returned by the clock literally. I see two choices. We can do, early on, what George described; if the year is less than, say 70, it must be the 21st century. Or, we can add a century switch to the front panel of the clock and pass it along to the Modcomp when the clock is read. I doubt the effort of a hardware change is warranted and presume we will do the kludge sometime in the next year or two.

Ken

----- end -----