

Chapter 7- Date Expiration Schemes

These schemes occur out of the blue without notice most of the time. Most apps that use this scheme are beta apps that are expected to be updated before too long or demos of games and such. There are only two ways to read the current date or time on a Macintosh, one is to use the `_GetDateTime` trap the other is to use the Macintosh Global variable memory address `$020C` at which the current number of seconds that has elapsed since Midnight, January 1st is stored. Basically both routines are extremely similar and would probably look something like this:

`_GetDateTime` (Protection Method)

```

    _GetDateTime          (Get current # of secs since 12am 01/01/04)
    CMPI.L expiredate,D1 (Compare the date with expiration date)
    BLE      notexpired   (If less than, then it hasn't expired)
    JSR      EXPIRED!     (Otherwise show the dialog and bomb out!)
```

Address `$020C` (Protection Method)

```

    CMPI.L expiredate,$020C (Compare the current date with expire date)
    BGT      EXPIRED        (If current date is > then expire date,
show dialog
                                & BOMB!)
    ----- (otherwise it hasn't expired and everything is
                                ok!)
```

That should have given you some idea what the protection schemes look like, they vary of course but that is what they will normally be like. The simple way to disarm these routines is to change the Branch Condition, for instance in the first protection scheme shown above it would be extremely easy to change the BLE to a BRA which will automatically branch to the "notexpired" portion of the program.

For the second routine has an easy way to eliminate the protection scheme. It would be to change the 'BGT EXPIRED' into two NOPs. Which will totally eliminate the check and fool the program into thinking everything is ok.