

---

---

## DATEPATCH

---

---

By: Bill van Melle  
(vanMelle.pa@Xerox.com)

DATEPATCH fixes some bugs and extends the functionality of the date functions DATE, GDATE and IDATE.

### Date Parsing

The date parser (IDATE) now handles all dates legal in RFC822 syntax (except the silly single-digit military time zones). In addition, it handles months spelled out, months abbreviated with a period, and ignores initial strings of the form "{letter}\*", assuming these to be specifying a day, as in "Monday, May 1, 1989". In addition to the official time zone specifications, it also recognizes any in the list TIME.ZONES, whose format has changed slightly:

TIME.ZONES

[Variable]

An association list whose elements are of the form (*offset regzone dstzone*), where *offset* is the number of hours west of GMT (note that this, unfortunately, is opposite in sign to the RFC822 standard, but is strictly an internal matter), *regzone* is a string specifying the time zone normally, and *dstzone* is a string specifying the zone when daylight savings time is in effect. If *dstzone* is omitted, then there is no representation for that zone in daylight savings time, and DATE is forced to use absolute syntax (e.g., +0400).

The initial value of TIME.ZONES is

```
( ( 8 "PST" "PDT" )
  ( 7 "MST" "MDT" )
  ( 6 "CST" "CDT" )
  ( 5 "EST" "EDT" )
  ( 0 "GMT" "BST" )
  ( 0 "UT" )
  (-1 "MET" "MET DST" )
  (-2 "EET" "EET DST" ) )
```

IDATE also accepts an optional argument DEFAULTTIME, which is interpreted as a number of seconds past midnight. If the date string does not contain a time, DEFAULTTIME is used; if DEFAULTTIME is NIL, IDATE returns NIL in this case (as it always has).

### Date Output

The date printers (DATE and GDATE) now produce appropriate time zones outside of the U.S. Given a choice of time zones, they take the first entry found in TIME.ZONES. In addition, they support a few more DATEFORMAT options:

MONTH.LONG

[DateFormat Option]

Provides for full names of months rather than the first three characters. For instance, (DATE (DATEFORMAT MONTH.LONG SPACES NO.TIME) ) might produce "20 February 87".

MONTH.LEADING

[DateFormat Option]

Causes the month to be produced as a word before the day and the day to be followed by a comma. For instance, (DATE (DATEFORMAT MONTH.LEADING MONTH.LONG YEAR.LONG NO.TIME)) might produce "February 20, 1987". MONTH.LEADING implies SPACES and inhibits NUMBER.OF.MONTH.

CIVILIAN.TIME

[DateFormat Option]

Specifies 12-hour time instead of 24-hour (military) time. For instance, (DATE (DATEFORMAT CIVILIAN.TIME NO.DATE NO.SECONDS)) might produce "11:34pm".

For completeness, listed below are all the DATEFORMAT options currently supported.

*Those affecting the date portion:*

NO.DATE	Omit the date portion (month, day, year, day of week).
NUMBER.OF.MONTH	Use a number for the month instead of spelling it.
MONTH.LONG	Spell the month out instead of abbreviating it.
MONTH.LEADING	Month before day, spelled out, comma after day.
YEAR.LONG	Use 4 digits for year instead of 2.
DAY.OF.WEEK	Include day of week (it appears at the end of the string, in parentheses).
DAY.SHORT	Use 3-letter abbreviation for day.
SLASHES	Separate parts of date with slashes instead of hyphens.
SPACES	Separate parts of date with spaces instead of hyphens.
NO.LEADING.SPACES	Omit leading spaces (default is a fixed format that always "lines up"). This also affects the hour when CIVILIAN.TIME is specified.

*Those affecting the time portion:*

NO.TIME	Omit the time portion (hour, minutes, seconds, zone).
NO.SECONDS	Omit seconds.
CIVILIAN.TIME	12-hour instead of 24-hour time.
TIME.ZONE	Include time zone specification.