

NAME

`rrdate` – print date in the revised Ravtaalian Calendar of Progress

SYNOPSIS

```
rrdate [ [ -gr ] mon date [ year ] ]
rrdate [ -c [ days ] ] [ -s days ] -y [ years [ days ] ... ]
```

DESCRIPTION

By default `rrdate`, like `date(1)`, prints the current date, but using the revised Ravtaalian Calendar of Progress, or *RRCP* for short, instead of the Gregorian (ei. "normal") calendar. It does not support any of the `date(1)` flags however, nor the UNIX notation for flexible reformatting of the default date, it will always be "%b %d %Y", that is "Month-abbr. Date Year". See *EXAMPLES* below however.

Also a complication with this calendar is that a yearly calendar scheme must be written before it can be used. The reason for this is that the RRCP follows a lunar year (12 months of 30 days), not a solar year. This means that a normal RRCP year is five or six days shorter than a solar year, to compensate for this a 13th intercalary month of arbitrary length is sometimes added. This calendar scheme must be mapped out before `rrdate` can be used, it cannot simply be "guessed".

The easiest way to make such a scheme, is to map out the years automatically with `rrdate -y`. This will assume that 1 March 1970 matches 1 Hajlmüian 1970 exactly (in other words that the RRCP year 1970 was perfectly synced with the Gregorian solar year 1970), that 1969 of RRCP did not have an intercalary month, and that the following years between 1970-2070 followed a sensible scheme, where an intercalary month of 30 days were introduced if the calendar was 10 days or more out of sync with the solar year. In other words, it will produce a very idealized and sensible lunar calendar scheme, although history has taught us that reality is seldom ideal. All of these assumptions can be adjusted however, see the *ADJUSTING CALANDAR SCHEME* section below.

It is also possible to specify an alternate date to convert to, either by providing a different Gregorian date which will be converted to a Ravtaalian one, or by providing a Ravtaalian date which will be converted to a Gregorian one. You can force `rrdate` to read a Ravtaalian or Gregorian date using the `-r` and `-g` flags, but normally there is no reason to do so. Only the month of "Feb" is ambiguous, since it can refer to the Gregorian month of February or the Ravtaalian month of Febramüian. In this case `rrdate` will default to the Gregorian month (if you write febra... on the other hand `rrdate` will understand that you mean Febramüian).

The following options are available, when specifying an alternate date:

- `-r` the date given is a Ravtaalian date
- `-g` the date given is a Gregorian date

ADJUSTING CALANDAR SCHEME

The following options are available, when you are mapping out the yearly scheme (these flags must be specified before `-y`)

- `-c` 1969 of RRCP did have an intercalary month, with the length of 30 days, unless you specify otherwise (space followed by number of days)
- `-s` The RRCP was *n* number of days out of sync with the solar year at the end of 1969. Negative numbers mean the year began too early, positive numbers too late. If an intercalary month was specified with the `-c` flag, these days will be in addition to the number of days specified with the `-s` flag.
- `years` You can prefix `-y` with a list of space separated numbers, that specify which years between 1970 and 2070 have intercalary months, prepend the years with the length of these intercalary months to set a different value than the default of 30 days.

REVISED CALENDAR

The revised Ravtaalian calendar has the same basic structure as the classic Ravtaalian calendar, 12 months, grouped in three blocks of four, where each commemorate a form of bad weather, a type of disease, an unpleasant insect and finally a furry miscreant. The months have 30 days however, not 28 as in the classic calendar, and the 13th month, celebrating ennui, is made into an intercalary month, that only appears in

seemingly random years by the Kings whim.

Spring Hajlmüan (from hajl, "hail")
 Plagamüan (from plaga, "plague")
 Mathamüan (from matha, "maggot")
 Wezlmüan (from wezlen, "weasel")

Summer Törthmüan (from törth, "drought")
 Tsimlmüan (from tsimel, "mildew")
 Klombarmüan (from klombar, "crane fly")
 Ratamüan (from rata, "rat")

Autumn/Winter Daachmüan (from daach, "fog")
 Febramüan (from febra, "fever")
 Grankarmüan (from grankar, "weevil")
 Aaslmüan (from aasel, "donkey")
 Lenklmüan (from linkwil, "tedium")

EXAMPLES

Just print a feasibly correct date:

```
; rrdate -y
; rrdate
```

Print the revised Ravtaalian date for February 23 2015

```
; rrdate Feb 23 2015
```

Print Gregorian date for the RRCP date Febramüan 23 2015

```
; rrdate -r Feb 23 2015
; rrdate febramüan 23 2015
```

Print RRCP date for today, but in the format "%a %b %d %T %Z %Y" (the default *date(1)* output on many systems)

```
; date='{date} rrdate | awk '{ print "'$date(1)'^",
$1, $2, "'$date(4)'^", "'$date(5)'^", $3
}' | sed 's/Wed/Woj/'
```

Print RRCP date for February 23 2015, but in the format "%Y-%m-%d" (ISO-8601 standard without time and timezone suffix)

```
; rrdate Feb 23 2015 | awk '
{ sub(/Aas/, "12", $1); print $3 "-" $1 "-" $2 }'
```

Map out a rather chaotic, and perhaps politically more plausible, lunar calendar scheme (where 1970 RRCP arrived 4 days too early):

```
; rrdate -s -4 -y 1981 31 1982 31 1987 31 1993 32 2004 2005 2007 2013 2019 31 2025 32 2031 31
2037 32 2061 2063 2065 2067 2069
```

FILES

\$HOME/plan9/lib/rrcp The yearly calendar scheme

It is possible to hack this file directly and create a truly unique calendar scheme. The last example above, for instance, will produce these lines in **\$HOME/plan9/lib/rrcp**:

```
4752000 1970 -4
35856000 1971 -9
66960000 1972 -14
...
1073088000 2004 -57 +days 30
```

1106784000 2005 -33 +days 30
1140480000 2006 -8
1171584000 2007 -13 +days 30
1205280000 2008 12

...

The fields are: the second the year starts, the year, how many days the year is out of sync with the solar year, and finally an intercalary month is indicated with +days, followed by the length of the 13th month. The script only uses the first two fields, the rest are put there to make the database more humanly readable. It makes it easy to check how good/bad your calendar scheme is. Ideally a lunar calendar should not be more than 15 days out of sync with the solar year (the provided example is a really bad lunar calendar).

BUGS

Converting today's week day into Jameld is easy, but finding the correct week day for any arbitrary date, is not. *Sorr date* doesn't even try. (see *cal(6)* however)

In true UNIX tradition *rrdate* naively assumes that time began 1 January 1970.

rrdate can only suggest a calendar scheme, the official RRCP calendar is defined by the Jameldic King.

The revised Ravtaalian Calendar of Progress is universally hated

SEE ALSO

date(1), *rdate(6)*, *rcal(6)*, *rcalendar(6)*, *rcron(6)*