ice_cube

John Crepezzi
Patch Media

What is ice_cube?

Ruby date recurrence.

Ready for DST,
Ready for TimeWithZone,
Fast

The first and last monday of april, every year

The first and _second to_ last monday of april, every year

Enter ice_cube

```
rule = Rule.yearly
    .day_of_week(:monday => [2, -1])
    .month_of_year(:april)
```

[Mon Apr 26 18:23:16 -0400 2010, Mon Apr 11 18:23:16 -0400 2011, Mon Apr 25 18:23:16 -0400 2011, Mon Apr 09 18:23:16 -0400 2012, Mon Apr 30 18:23:16 -0400 2012, Mon Apr 08 18:23:16 -0400 2013, Mon Apr 29 18:23:16 -0400 2013, Mon Apr 14 18:23:16 -0400 2014, Mon Apr 28 18:23:16 -0400 2014, Mon Apr 13 18:23:16 -0400 2015]

0.010735 s

iCalendar

Implements all rule/validation types present in the RFC

daily, weekly, monthly, yearly, secondly, minutely, hourly,

day_of_week, day, month_of_year, day_of_year, second_of_minute, minute_of_hour, hour_of_day, day_of_month

until, count

You can make some pretty complex combinations

Presidential Election Day
```
Rule.yearly(4)
    .month_of_year(:november)
    .day(:tuesday)
    .day_of_month(2, 3, 4, 5, 6, 7, 8)
```

ice_cube's spec suite implements <u>every</u> example from the RFC

How to use a schedule

```
schedule = Schedule.new(Time.now)
schedule.add_recurrence_rule Rule.daily

# first 10 occurrences
schedule.first(10)

# occurs on a certain day?
schedule.occurs_on(Time.local(2010, 7, 2))

# occurrences until a certain time
schedule.occurrences(Time.local(2011, 9, 2))

# all occurrences
schedule.all_occurrences
```

In your application

Rule.daily.day(:tuesday).day_of_week(:monday => [1, -2])

# to_ical
FREQ=DAILY;BYDAY=TU;BYDAY=-2MO,1MO

to_s (international coming soon)
Daily on Tuesdays on the 2nd to last and 1st Mondays

to_yaml

to_hash

Questions?

http://github.com/seejohnrun/ice_cube
http://seejohnrun.github.com/ice_cube/
gem install ice_cube