treacle Documentation

Release 0.1.2

Caramel

Feb 25, 2019

Contents

treacle is an AGI (Asterisk Gateway Interface) script which handles public holidays and opening hours in different offices with different timezones and different public holidays.

This allows conditional call routing in Asterisk, and also allows one system to route calls for different time zones and different offices.

This means if you have callers in Adelaide and callers in Melbourne, you can give each a different number, and their calls will be routed to voicemail after hours in their timezone, and not available on public holidays in their timezone.

- Treacle on PyPI (releases)
- GitHub repository
- On-line Documentation

Contents:

CHAPTER 1

Installing treacle

treacle requires either Python 2.6 or 2.7. Python 3 isn't presently supported.

1.1 Installing stable version with pip

If you don't already have pip installed, you'll need to install it with `easy_install` first:

```
# easy_install pip
```

Then you can install the current stable version of treacle:

```
# pip install treacle
```

Then you're done, and treacle will be available in your PATH.

1.2 Installing development version from git

You can install the current development version of the code from the `git` repository:

```
# git clone -b develop https://github.com/Caramel/treacle.git
# cd treacle
# pip install -r requirements.txt
# python setup.py install
```

CHAPTER 2

Configuring treacle

Treacle, by default, looks for it's configuration file in `/etc/treacle/treacle.ini`.

You can override this with the `-c` option to treacle, which is covered in a later section. For the purposes of this document, I'll assume you're working with treacle's configuration at that location.

This file uses the same format as Python's `ConfigParser` module, which is similar to Windows INI format.

The first thing that treacle reads when determining holidays is the `[DEFAULT]` section. Options in there are then overridden by the other sections which represent offices (in the example below, `adel`, `melb` and `nuri`).

2.1 Example configuration

```
[DEFAULT]
holidays = /etc/treacle/holidays/au_holidays.ics

[adel]
holidays_sa = /etc/treacle/holidays/sa_holidays.ics

tz = Australia/Adelaide
hours1 = Mon,Tue,Wed,Thu,Fri@08:30-18:00

[melb]
holidays_vic = /etc/treacle/holidays/vic_holidays.ics

tz = Australia/Melbourne
hours1 = Mon,Tue,Wed,Thu,Fri@08:00-16:00

[nuri]
holidays_sa = /etc/treacle/holidays/sa_holidays.ics
holidays_nuri = /etc/treacle/holiday/nuri_holidays.ics

tz = Australia/Adelaide
```

(continues on next page)

(continued from previous page)

```
hours1 = Mon,Tue,Wed,Thu,Fri@07:00-19:00
hours2 = Sat@12:00-15:00
```

The `adel` office will use the holidays files `au_holidays.ics` and `sa_holidays.ics`, uses the `Australia/Adelaide` time zone, and is open from Monday to Friday, 08:30 to 18:00 local time.

The `melb` office will use the holidays files `au_holidays.ics` and `vic_holidays.ics`, uses the `Australia/Melbourne` time zone, and is open from Monday to Friday, 08:00 to 16:00 local time.

The `nuri` office will use the holidays files `au_holidays.ics`, `sa_holidays.ics` and `nuri_holidays.ics`. It uses the `Australia/Adelaide` time zone, and is open from Monday to Friday, 07:00 to 19:00 and on Saturdays from 12:00 to 15:00.

2.2 Options reference

The options available are as follows:

tz An Olson (UNIX) time zone name representing the time zone at the location. If this is not specified, UTC (Coordinated Universal Time) is assumed.

For example, `Australia/Adelaide` or `Australia/Melbourne`.

Changed in version 0.1.2: As of v0.1.2, a default time zone of UTC is assumed if no default or office timezone is specified. In previous versions of treacle, this was an invalid configuration and would cause an error.

holidays* A path to an iCalendar file where public holidays that apply to this office is located.

This should be specified as an absolute path, as relative paths are treated relative to the working directory that treacle is run from, which can lead to unpredictable results.

Option names starting with `holidays` are allowed, which will allow you to specify multiple holiday files for a single location. In the example above, this is indicated with multiple option names (`holidays`, `holidays_sa`, `holidays_vic`).

hours* The opening hours for the office, specified in the time zone of the office.

It is specified in the following format:

```
Mon,Tue,Wed,Thu,Fri@09:00-17:00
```

Where:

- The days that these hours apply to and the hours are separated by an `@` symbol.
- The days of the week are specified in your locale in either long (`Monday`) or short (`Mon`) form, with multiple days separated by commas `,`.
- Opening hours are specified in 24-hour format, with the start and end time separated by a dash `-`.

Multiple sets of hours may be specified by providing multiple options that start with the string `hours`. For example, `hours1`, `hours2`, etc.

Localised day names are generated using the `%a` and `%A` strftime formatting codes. If localised day names are not desired, set the environment variables `LANG=C LC_TIME=C` when running treacle.

CHAPTER 3

Using treacle in Asterisk dialplans

Treacle can be used inside of Asterisk dialplans as an AGI (Asterisk Gateway Interface) script, with the option `-a`.

For example:

```
[my_dialplan]
exten => s,1,AGI(/usr/local/bin/treacle,-a,-o,adel)

; Print returned output from treacle and make decision
exten => s,n,NoOp(Business Hours = ${CARHOURS})
exten => s,n,GotoIf(${CARHOURS}?open:closed)

; Business hours
exten => s,n(open),Queue(myqueue,t,,,30)

; Fall through if queue fails

; Not available, and failure mode for queue
exten => s,n(closed),Voicemail(100,u)

; Hangup when done.
exten => s,n,Hangup
```

The dialplan variable `CARHOURS` will be set by the AGI script on completion. This will be set to `0` if it is not business hours for the location, and `1` if it is business hours for the location.

In this example, it will send the caller into the queue `myqueue` if it is currently business hours, and if they have been in the queue for more than 30 seconds or it is not business hours, direct them to the voicemail box `100` with the `unavailable` state.

You can also check if any office is in business hours:

```
[my_dialplan]
exten => s,1,AGI(/usr/local/bin/treacle,-a,-A)
```

And set a different dialplan variable:

```
[my_dialplan]
exten => s,1,AGI(/usr/local/bin/treacle,-a,MYVAR,-o,melb)
exten => s,n,NoOp(Business Hours = ${MYVAR})
```

Or use a different configuration file:

```
[my_dialplan]
exten => s,1,AGI(/usr/local/bin/treacle,-c,/etc/treacle/examplecorp.ini,-o,gamb)
```

Note: On some systems (such as Red Hat and CentOS), `distutils` installs treacle to `/usr/bin`, not `/usr/local/bin`. Adjust this accordingly for your system.

CHAPTER 4

Using treacle in shell scripts (standalone)

Treacle be used outside of Asterisk, with the option `-s`:

```
treacle -s -o adel && echo "Adelaide Office Open"
treacle -s -o melb || echo "Melbourne Office Closed"
```

This will print `Adelaide Office Open` if the Adelaide office is currently open, and `Melbourne Office Closed` if the Melbourne office is closed.

The exit code `0` will be given if the office is open, and the exit code `1` will be given if the office is closed.

Like in AGI, you can use this to find if any office is open:

```
treacle -s -A && echo "There is an office open"
```

This will print `There is an office open` if any office is currently open.

CHAPTER 5

Data sources

Data sources are given as iCalendar files. You can create these yourself, or download them from various sources online.

Recurring events are not supported by this software.

All events inside of the iCalendar file are read using the icalendar library, so will inherit it's limitations.

Events that don't have a direct, Olson timezone name on them won't carry timezone information. In this case, `treacle` will make the events be interpreted in the timezone for the office being read (this includes those inherited from defaults). If any calendar entry lands within a daylight savings transition point, if the time is ambiguous or non-existent, it is assumed to be within "standard" time.

This applies even if there is `VTIMEZONE` information in the calendar file. This is a known issue in icalendar.

5.1 Australian Holiday Data

5.1.1 Official government data sources

Some states release official iCalendar feeds of public holidays:

- Australian Capital Territory
- South Australia
- Victoria

5.1.2 Scraping Apple for public holidays

For Australian users, a tool is included to scrape the Apple iCal public holiday feeds. In order to use this after you have installed the software:

```
$ treacle_scrape
```

This will produce the following files in the current directory:

- `au_holidays.ics` - Contains holidays specific to no state (national holidays)
- `act_holidays.ics` - Contains holidays specific to ACT
- …and so on for `nsw`, `nt`, `qld`, `sa`, `tas`, `vic` and `wa`.

This script will blindly overwrite these files in the directory, so be careful!

Note: This doesn't take into account a number of quirks of public holidays in various states.

For example, in South Australia, Christmas Eve and New Years Eve are public holidays for part of the day. These aren't listed at all.

In Tasmania, there are holidays that only apply for certain cities. Apple appears to only provide information about holidays in Hobart.

5.2 Special office closures

If you wish to have a special calendar with office closures in it, you can create another iCalendar file with these closures in it, and add it as a holiday source.

Because this software doesn't run as a daemon, you can update the calendar file whenever you like, and this will apply to all calls recieved after this point. For example:

```
[DEFAULT]
holidays = /etc/treacle/holidays/au_holidays.ics
holidays_closed = /etc/treacle/holidays/exampleco_cloures.ics

[adel]
holidays_sa = /etc/treacle/holidays/sa_holidays.ics
hours = Mon,Tue,Wed,Thu,Fri@09:00-17:30
```

The holidays ICS files can be updated from an external source, and `exampleco_closures.ics` would contain information about your office closures.

treacle supports part-of-day events (as of v0.1.2), so you can add an event that would close your office early for the day, or start late. You can also make these apply to only certain offices.

CHAPTER 6

`treacle` module

Treacle can also be used by other Python programs in order to provide similar after-hours functionality.

6.1 `treacle.Treacle` class

class `treacle.`**`Treacle`**(*config*, *config_as_dict=False*)

Class for calculating office hours in multiple offices.

Parameters

- **`config`** (*`file or dict`*) – Configuration source to use. This uses syntax defined in *Configuring treacle*.
- **`config_as_dict`** (*`boot`*) – Treat the parameter `config` as a `dict` rather than a `file`, if True. This is useful when passing in configuration from a non-file source, such as for unit testing.

`in_hours`(*office=None*, *when=None*)

Finds if it is business hours in the given office.

Parameters

- **`office`** (*`str or None`*) – Office ID to look up, or None to check if any office is in business hours.
- **`when`** (*`datetime.datetime`*) – When to check the office is open, or None for now.

Returns True if it is business hours, False otherwise.

Return type bool

Raises **`KeyError`** – If the office is unknown.

6.2 `treacle.Office` class

class `treacle.`**`Office`**(*config*, *section*)

Represents an office configuration for Treacle.

This class should only be instantiated by *`Treacle`*, not by user applications.

Parameters

- **`config`** (*`configparser.ConfigParser`*) – Configuration object to use.
- **`section`** (*`str`*) – Name of the section to read this office's configuration from.

`in_hours`(*when*)

Find if the given `datetime` is in business hours for this office.

Parameters **`when`** (*`datetime.datetime`*) – The time to check

Returns True if the given time is in business hours for the office, False otherwise.

Return type bool

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Index