

Re: convert time string in UTC to time in local time

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2007-03/msg01374.html>

- *From:* "Paul Boddie" <paul@xxxxxxxxxxxxxx>
 - *Date:* 10 Mar 2007 16:57:44 -0800
-

davelist@xxxxxxx wrote:

I'm guessing there is an easy way to do this but I keep going around in circles in the documentation.

I have a time stamp that looks like this (corresponding to UTC time):

```
start_time = '2007-03-13T15:00:00Z'
```

I want to convert it to my local time.

```
start_time = time.mktime(  
time.strptime(start_time, '%Y-%m-%dT%H:%M:00Z')  
)
```

Here, `strptime` in this case will probably produce a time whose daylight saving time (DST) status will be undecided, although I've done some tests with `glibc` which suggests that the native (unused by Python) `strptime` produces non-DST-affected times for both the given date and for a date one month later in my time zone (contrary to what one might expect). Then, `mktime` is called, but this may apply various time zone magic to treat the supplied time as a local time, not a GMT/UTC time – this seems to be an artifact of the way `mktime` is defined and implemented in various standards.

Do this to see what I'm talking about:

```
time.mktime(time.gmtime(0))
```

Unless you have GMT/UTC as your time zone, the above may well not produce zero as its result. Similar tests in C behave identically.

```
start_time -= time.timezone
```

So, here you've applied a time shift in order to hopefully go from GMT/UTC to local time. Given my observations of `mktime`, I'm inclined to

Re: convert time string in UTC to time in local time

believe that you've time shifted beyond local time, but the principle seems reasonable. See below for more discussion.

This was working fine now, but if I do it for a date next week (such as March 13th in the above example), it breaks because my local time moves to daylight savings time this weekend. So my time is now off by an hour. I'm guessing if I try this next week it will work okay because `time.timezone` will be give a different value next week – is that correct?

The value of `time.timezone` should remain the same. In your time tuple/structure (produced by `strptime`) there's a field called `tm_isdst` (it's the value in the last position) which governs when DST applies. If DST applies you need to use `time.altzone` instead of `time.timezone` to guide conversions between local time and GMT/UTC.