Sometimes you need to change the timezone files to cope with changes to your local legal environment, such as:

- in Australia when a state government decides to shift the start or end of daylight saving (Victoria's adjustment for the Melbourne Commonwealth Games and Western Australia's temporary adoption in late 2006), or
- in the US when the state of Indiana decided in 2006 to start using Daylight Saving Time for the first time in many decades, and at the same time some Indiana counties switched time zones (from Eastern to Central).

Background: Debian includes timezone files in the */usr/share/zoneinfo* directory tree, which is provided by the *tzdata* package for Etch and later. (For Sarge and earlier the files are part of the *libc6*). These packages may be updated from time to time:

- In Etch, new versions of the *tzdata* package were released periodically when new versions of the upstream data files became available (at least until the release freeze in late 2006).
- In Sarge, the libc6 package version 2.3.2.ds1-22sarge3 updated the timezone files to the tzdata2006b upstream release. This libc6 package version was included in the Sarge "r2" update, released on April 19, 2006.
- In Woody, the latest glibc6 2.2.5-11.8 package has not changed and so does not include recent timezone changes.

To make a change ahead of the change being made by the Debian project, follow this process:

- check that the system is configured for the correct timezone,
- run *zdump* to make sure the change is needed,
- obtain an updated timezone source file, by download or patch,
- run *zic* to make the change,
- run *zdump* again to make sure the change is effective,
- verify by other means.

More detail below, using Sydney Australia as the example. Substitute the timezone name and the file name for your local environment.

Check Configured Timezone

To see or change what timezone your Debian system is configured for,

```
# dpkg-reconfigure tzdata
```

What does this do? It changes /etc/localtime. Background information follows.

In Debian releases Etch and later, /etc/localtime is a copy of the original data file. Check the contents of /etc/timezone to see the name of the timezone. If the system is configured normally, you should find that the zoneinfo file referenced by this name is identical to /etc/localtime. For example, you can verify this by running:

```
$ diff -s /etc/localtime /usr/share/zoneinfo/`cat /etc/timezon
Files /etc/localtime and /usr/share/zoneinfo/America/New_York
```

The *tzconfig* command updates both /etc/localtime and /etc/timezone.

In releases Sarge and earlier, /etc/localtime is a link to a file. Check where /etc/localtime links to:

```
# ls -l /etc/localtime
lrwxrwxrwx    1 root     root           48 Mar 31 11:19 /etc/l
```

and use the path following the "/usr/share/zoneinfo/" prefix.

Check If Needed

The *zdump* program dumps the timezone table. Use it to check the dates for the daylight savings change. For example this system has old Australian

rules, showing a change from daylight saving on 26th March:

```
# zdump -c 2007 -v Australia/Sydney|grep 2006
Australia/Sydney  Sat Mar 25 15:59:59 2006 UTC = Sun Mar 26 02
Australia/Sydney  Sat Mar 25 16:00:00 2006 UTC = Sun Mar 26 02
Australia/Sydney  Sat Oct 28 15:59:59 2006 UTC = Sun Oct 29 01
Australia/Sydney  Sat Oct 28 16:00:00 2006 UTC = Sun Oct 29 03
```

Get Timezone File

Obtain a copy of the *australasia* or country-specific timezone file a recent tzdata package, or from the upstream site.

To get the *australasia* file from the *tzdata* sources, configure your *sources.list* with *deb-src* lines for your current distribution version, fetch the source, unpack it, and then edit the file. For example on a system tracking Lenny (version numbers may vary):

```
# mkdir /tmp/tzdata
# cd /tmp/tzdata
# apt-get install dpkg-dev
# apt-get source tzdata
# cd tzdata-2008e
# tar xfz tzdata2008e.tar.gz
# more australasia
```

To bypass Debian and get the files from where the Debian project got them from:

```
% mkdir /tmp/tzdata
% cd /tmp/tzdata
% wget 'ftp://elsie.nci.nih.gov/pub/tzdata*.tar.gz'
% tar xfz tzdata*.tar.gz
% more australasia
```

You should now have all the timezone files.

If you are using the file from the source package without the change, then change the file adding the new lines. For example these lines are in the corrected file for the early 2006 change in *australasia*:

```
...
Rule    AN      2001    max     -       Oct     lastSun 2:00s
Rule    AN      2006    only    -       Apr     Sun>=1  2:00s
Rule    AN      2007    max     -       Mar     lastSun 2:00s
...
```

Pretest Change

This is an optional step. It tests the source file without changing the system configuration. It can be done as an ordinary user.

```
$ mkdir test-tz
$ /usr/sbin/zic -d test-tz australasia
$ TZ=`pwd`/test-tz/Australia/Sydney date --date="2006-04-01 16
Sun Apr  2 02:00:10 EST 2006
```

This test works by compiling the timezone source file into a directory, then asking the *date* command to test translating a UTC time into localtime near the expected discontinuity.

Commit Change

The *zic* program changes the timezone table. Give it the name of the *australasia* or country-specific source file you changed. For example:

```
# /usr/sbin/zic australasia
```

Then reconfigure the tzdata package, which will update /etc/localtime:

```
# dpkg-reconfigure tzdata
```

Verification Test 1: Dump Timezone Rules

Use the *zdump* program again to confirm the change. For example this system has the change properly applied, showing a change from daylight saving on 2nd April:

```
# zdump -c 2007 -v Australia/Sydney|grep 2006
Australia/Sydney  Sat Apr  1 15:59:59 2006 UTC = Sun Apr  2 02
Australia/Sydney  Sat Apr  1 16:00:00 2006 UTC = Sun Apr  2 02
Australia/Sydney  Sat Oct 28 15:59:59 2006 UTC = Sun Oct 29 01
Australia/Sydney  Sat Oct 28 16:00:00 2006 UTC = Sun Oct 29 03
```

Verification Test 2: Translate a UTC Time

You can use the *--date* option to the *date* command see how the system will interpret a particular time. This method does not change the system clock.

```
$ date --date="2006-04-01 15:59:50 utc"
Sun Apr  2 02:59:50 EST 2006
$ date --date="2006-04-01 16:00:10 utc"
Sun Apr  2 02:00:10 EST 2006
```

Verification Test 3: Observe Rollover

It is possible to change your system's time so as to demonstrate the fix is good. There are side-effects of changing the time, such as distortion of file dates, or interoperability problems with other systems nearby, so it's only a good idea on a test system.

To do the test, use the *date --utc* command to change the time to just before the first discontinuity identified by *zdump*, then use the *date* command to show the local time, over the discontinuity. For example:

```
# date --utc 040115592006.50 ; date ; sleep 20 ; date
Sun Apr  2 02:59:50 EST 2006
Sun Apr  2 02:00:10 EST 2006
```

Test Failure Scenarios

- setting the time in UTC format using the *--utc* flag is helpful, because setting the time to 02:59:50 would be ambiguous and the *date* command takes it to mean the second 02:59:50 that morning rather than the first, and the rollover isn't seen to happen.
- if you have NTP enabled, changing the system time in the final test does not work so well, turn it off first.
- if the rollover or translation test fails, yet *zdump* looks fine, perhaps your system has an */etc/localtime* file that is a copy of the zoneinfo file rather than a symlink to it. Debian GNU/Linux in Sarge and earlier uses a symlink, but some other Linux distributions use a copy of the file.

Restarting Daemons and Long-Running Programs

After the zoneinfo files are updated, you may need to restart daemons and other long-running programs to get them to use the new zone information. Examples of such programs include *apache*, *bind*, *cron*, *fetchmail -d*, *inetd*, *mailman*, *sendmail*, and *sysklogd*. A common symptom of this problem is seeing incorrect timestamps mixed in with the correct timestamps in your log files (e.g. */var/log/syslog*). Even interactive programs like "mutt" may continue to use the old timezone information until they are restarted.

Binary Portability

The compiled zoneinfo file appears to be platform and architecture independent, so if you have multiple machines to update you should be able to move the file as-is without running *zic* again. Similarly, you can just to pull updated zoneinfo files from a machine running a different Debian release, or even download the latest tzdata package from unstable, extract the zoneinfo files from within the .deb file, and install them into the /usr/share /zoneinfo directory tree.

References

- ~~Closed in glibc/2.3.6-1: #345479: Summer time in 2006 Australia/NSW is incorrect...: Bug #345479~~ covers the 2006 timezone changes for Australia
- http://www.twinsun.com/tz/tz-link.htm the upstream mailing list for the time zone files
- ftp://elsie.nci.nih.gov/pub/ the upstream source for the time zone files
- http://www.macosxhints.com/article.php?story=20060117170839977&lsrc=osxh also describes the process
- http://www.timetemperature.com/tzus/indiana_time_zone_map.shtml gives more information on the 2006 time zone changes in Indiana