
Article

[Mark Bolinsky](#) · Feb 3, 2016 2m read

Linux TZ environment variable not being set and the impact to Caché

During recent large scale benchmarking activities, we were seeing excessive %sys CPU time that negatively impacted scaling of the application.

Problem

We have found that a lot of the time was spent in the localtime() system call due to the TZ environment variable not being set. A simple test routine was created to confirm the observation, and the elapse time differences and CPU resources needed with TZ set versus TZ not set were astonishing. It was discovered that the inherit use of stat() system calls to /etc/localtime from localtime() are very expensive when TZ is not set.

Recommendation

InterSystems highly recommends confirming in any Linux installation whether x86 or Linux on Power that the TZ environment variable is set appropriately for optimal performance. For more details see: "man tzset".

The current Caché 2016.1 Field Test contains optimizations in regards to date and time functions, and initial testing indicates a profound improvement. Below are sample outputs from testing the new internal function calls in a loop with Linux on Power in both cases with TZ *not* being set:

Before Caché 2016.1 FT:

```
real 0m22.60s
```

```
user 0m1.64s
```

```
sys 0m20.89s
```

With Caché 2016.1 FT:

```
real 0m0.40s
```

```
user 0m0.37s
```

```
sys 0m0.00s
```

Please post comments with any experiences you may have had with your application regarding TZ environment variable and also the impact of Caché 2016.1 Field Test has with or without TZ being set.

[#Best Practices](#) [#InterSystems Business Solutions and Architectures](#) [#System Administration](#) [#Caché](#)

Source

URL: <https://community.intersystems.com/post/linux-tz-environment-variable-not-being-set-and-impact-cach%C3%A9>
