

# Converting UNIX dates to Excel

And dealing with Microsoft assuming dates/times are in the local timezone...

A UNIX date in seconds can be converted to a more readable form like this on (say) a Linux machine:

```
date --date='@2147483647'
```

```
Tue Jan 19 03:14:07 UTC 2038
```

On macOS:

```
date -r 2147483647
```

shows:

```
Tue 19 Jan 2038 03:14:07 GMT
```

In a shell such as bash the result can be shown in the London timezone like this:

```
TZ="Europe/London" date --date='@2147483647'
```

```
Tue Jan 19 03:14:07 GMT 2038
```

This can be then rendered in an Excel-friendly-ish and unambiguous YYYY/mm/dd HH:MM:SS form as follows:

```
TZ='Europe/London' date '--date=@2147483647' '+%Y/%m/%d %H:%M:%S'
```

or from another shell:

```
bash -c "TZ='Europe/London' date '--date=@2147483647' '+%Y/%m/%d %H:%M:%S'"
```

or on macOS:

```
bash -c 'TZ=Europe/London date -r 2147483647 "+%Y/%m/%d %H:%M:%S"'
```

To yield:

```
2038/01/19 03:14:07
```

To convert a whole file of the form:

```
XXX,received_timestamp,device_timestamp,energy,temperature
XXX,1470009880,1470009600,5876.5,11
```

preserving the device\_timestamp,energy,temperature fields only, and with the device timestamp converted to a more Excel-friendly form, on macOS in bash one might use:

```
awk '{cmd="date -r "$3" \"+%Y/%m/%d %H:%M:%S\""; cmd | getline var; cmd |
getline discard; close(cmd); print var "," $4 "," $5 ; }' < input.csv > output.csv
```

yielding output of the form:

```
2016/08/01 11:15:00,1447.58,14
2016/08/01 11:30:00,1447.58,14
2016/08/01 11:45:00,1447.58,14
2016/08/01 12:00:00,1447.58,14
2016/08/01 12:15:00,1447.58,14
2016/08/01 12:30:00,1447.58,15
2016/08/01 12:45:00,1447.58,14
```

However, it may well be necessary to use a DD/MM/YYYY date format, which will be misinterpreted if a user has US settings rather than UK for example:

```
awk '{cmd="date -r "$3" \"+%d/%m/%Y %H:%M:%S\""; cmd | getline var; cmd |
getline discard; close(cmd); print var "," $4 "," $5 ; }' < input.csv > output.csv
```

yielding output of the form:

```
01/08/2016 11:15:00,1447.58,14
01/08/2016 11:30:00,1447.58,14
01/08/2016 11:45:00,1447.58,14
01/08/2016 12:00:00,1447.58,14
01/08/2016 12:15:00,1447.58,14
01/08/2016 12:30:00,1447.58,15
01/08/2016 12:45:00,1447.58,14
```