



# ezLaborManager®

## Why Don't My Totals Agree?

### *Introduction*

This document has been assembled in response to a number of clients implementing ezLaborManager and noticing discrepancies between totals generated by their payroll system and totals generated by the reporting or Timecard functions of ezLaborManager. In all such circumstances, the totals are in fact correct, but the effect of rounding minutes worked to hours and hundredths of hours has created confusion when the accumulated rounding error causes the sum of two values to be smaller than the sum of the displayed individual totals.

In most reported cases, the confusion has been caused by a human being mentally summing the individual values and assuming that the displayed amount is in fact the actual amount used in the calculation. In fact, the displayed amount is a rounded value showing the two-digit decimal precision of the actual number, and the actual number is used in calculating any subsequent totals.

## Accumulation Example

Consider two employees, Romeo and Juliet. Both employees work on Monday, and both employees work for exactly 7 hours and 53 minutes.

				Interval Total		Cumulative Total	
	In	Out	Dept	HH:MM	HH.hh	HH:MM	HH.hh
<b>Juliet</b>	12:00	07:53	100	07:53	07.88	07:53	07.88
<b>Romeo</b>	12:00	06:52	300	06:51	06.87	06:51	06.87
	06:52	07:53	200	01:01	01.02	07:53	07.88

Romeo worked in two different accounts, and therefore has two punch pairs and two independent intervals. But they worked the same number of total hours, and this is reflected in the totals.

However, if you add the sums of the two hours and hundredths totals for Romeo, you will get a cumulative hours and hundredths total of 7.89, not 7.88. And this is what many managers are doing. In fact, the contributing totals are not 6.87 and 1.02, but 6.8666 and 1.0166. Adding these actual numbers and rounding to the nearest hundredth does in fact yield the ezLaborManager total, demonstrating that it is correct. The human, in accumulating the rounding error of the individual values, generates the different and incorrect result.

Why can't we simply accumulate the error in totals calculations and not confuse people? Well, then we'd have to explain why Romeo earned .01 hours more for working exactly the same amount of time!

A problem is that the payroll export will typically create individual output to each account worked, which when summed together will accumulate the rounding error. This will in fact result in Romeo getting paid more, because there is not sufficient data granularity at the payroll side to correct the rounding error.

## Minutes to Hundredths Conversion the ezLaborManager Way

The table below contains the conversion values for minutes amounts that are part of an elapsed total. ezLaborManager will never provide a totals amount that does not equal one of the Rounded Hundredths values included in the table below.

Minutes Computed	Rounded Hundredths	Actual Hundredths	Minutes Computed	Rounded Hundredths	Actual Hundredths
:00	0.00	0.000000	:30	0.50	0.500000
:01	0.02	0.016667	:31	0.52	0.516667
:02	0.03	0.033333	:32	0.53	0.533333
:03	0.05	0.050000	:33	0.55	0.550000
:04	0.07	0.066667	:34	0.57	0.566667
:05	0.08	0.083333	:35	0.58	0.583333
:06	0.10	0.100000	:36	0.60	0.600000
:07	0.12	0.116667	:37	0.62	0.616667
:08	0.13	0.133333	:38	0.63	0.633333
:09	0.15	0.150000	:39	0.65	0.650000
:10	0.17	0.166667	:40	0.67	0.666667
:11	0.18	0.183333	:41	0.68	0.683333
:12	0.20	0.200000	:42	0.70	0.700000
:13	0.22	0.216667	:43	0.72	0.716667
:14	0.23	0.233333	:44	0.73	0.733333
:15	0.25	0.250000	:45	0.75	0.750000
:16	0.27	0.266667	:46	0.77	0.766667
:17	0.28	0.283333	:47	0.78	0.783333
:18	0.30	0.300000	:48	0.80	0.800000
:19	0.32	0.316667	:49	0.82	0.816667
:20	0.33	0.333333	:50	0.83	0.833333
:21	0.35	0.350000	:51	0.85	0.850000
:22	0.37	0.366667	:52	0.87	0.866667
:23	0.38	0.383333	:53	0.88	0.883333
:24	0.40	0.400000	:54	0.90	0.900000
:25	0.42	0.416667	:55	0.92	0.916667
:26	0.43	0.433333	:56	0.93	0.933333
:27	0.45	0.450000	:57	0.95	0.950000
:28	0.47	0.466667	:58	0.97	0.966667
:29	0.48	0.483333	:59	0.98	0.983333