TimeFlies – A Tool for Time Keeping

Jörg Bullmann

September 27, 2012

Contents

1 What is *TimeFlies* About?

Do you need to account for the time you spend at work? What project or work package have you been working on? Do you want to keep track of the hours you work? How much leave have you got left for this year?

Do you keep a daily work log containing things you did, problems you solved, some kind of to do list?

Do you want to make an estimate of effort for a project or work package? Would you like to break down those things into smaller items and possibly break down those again too?

TimeFlies can help you with this.

2 Tutorial by Example

In this section we will look at a number of use cases. All *TimeFlies* data is kept in plain text files. So all your data is always easily accessible to you and the format itself is quite human-readable. Moreover it can easily be version controlled.

2.1 Recording Time

To record your work time keep a work log file with `day` lines specifying the dates and in and out times telling when you arrived at work and when you left. The times can be given in decimals or in hours and minutes:

```
day 2012-09-19 8:30 17:15
day 2012-09-18 8.75 17.75
day 2012-09-17 8 18
```

Do you need to account for breaks you are taking? Use `off` instructions to state periods of time in a day of work during which you were not actually working:

```
day 2012-09-19 8.5 17.25, off 0.5
day 2012-09-18 8.75 17.75, off 0.75
day 2012-09-17 8 18, off 0.5, off 0.25
```

This last file is equivalent to the following:

```
day 2012-09-19 8.5 17.25
off 0.5
day 2012-09-18 8.75 17.75
off 0.75
day 2012-09-17 8 18
off 0.5
off 0.25
```

This last example illustrates the notion of a `day`-*block*: a `day`-*block* extends from one `day`-keyword to the next and everything inside this day block is part of that day.

The days in the file do not need to be listed chronologically. You could e.g. list the days in reverse order so that the present is always at the top of the file.

If you want to mask out part of your log temporarily you can use the # source comment marker. *TimeFlies* ignores the # and everything following it until the end of line. It works just the same as e.g. a Python comment.

```
day 2012-09-19 8.5 17.25
off 0.5

# day 2012-09-18 8.75 17.75
# off 0.75

day 2012-09-17 8 18
off 0.5
off 0.25
```

2.2 Keeping Notes in the Log

Do you want to keep notes about your work in the same place as you keep the time information? Use *log comment* lines like in this file:

```
day 2012-09-19 8.5 17.25
; updated regression tests
off 0.5
; fixed build scripts

day 2012-09-18 8.75 17.75
; wrote unit test to reproduce problem report 2012-0098
; fixed problem report 2012-0098
off 0.75
; added HTML output option to object dumper
; discussed implications of Java 1.7 rollout

day 2012-09-17 8 18
; weekly team meeting
off 0.5
; monthly quality task force
off 0.25
; code review: server side includes
```

A log comment line starts with a semicolon and one or more space characters. All text following these characters until the end of the line (or until a #

source comment marker) with trailing spaces removed constitute the recorded log comment.

Now what can you do with such a file? Assume the above work log file's name is `work-log.fly`, then option `-t` tells *TimeFlies* to calculate your work times.

```
> timeflies.pl -t work-log.fly
Time at work overview (all):
2012-09-17, Mon:    9.25 worked, ----.-- leave, ----.-- sick
2012-09-18, Tue:    8.25 worked, ----.-- leave, ----.-- sick
2012-09-19, Wed:    8.25 worked, ----.-- leave, ----.-- sick
   week 2012-38:   25.75 worked, ----.-- leave, ----.-- sick
  month 2012-09:   25.75 worked, ----.-- leave, ----.-- sick
          total:   25.75 worked, ----.-- leave, ----.-- sick
```

To include the log comments in this output, use the `-C` option:

```
> timeflies.pl -t -C work-log.fly
Time at work overview (all):
2012-09-17, Mon:    9.25 worked, ----.-- leave, ----.-- sick
              ; weekly team meeting
              ; monthly quality task force
              ; code review: server side includes
2012-09-18, Tue:    8.25 worked, ----.-- leave, ----.-- sick
              ; wrote unit test to reproduce problem report 2012-0098
              ; fixed problem report 2012-0098
              ; added HTML output option to object dumper
              ; discussed implications of Java 1.7 rollout
2012-09-19, Wed:    8.25 worked, ----.-- leave, ----.-- sick
              ; updated regression tests
              ; fixed build scripts
   week 2012-38:   25.75 worked, ----.-- leave, ----.-- sick
  month 2012-09:   25.75 worked, ----.-- leave, ----.-- sick
          total:   25.75 worked, ----.-- leave, ----.-- sick
```

2.3 Time Summaries

Here's a longer example where you can see the use of weekly and montly summaries. Also a few days of annual leave are inserted using the `leave-days` instruction.

```
day 2012-08-23 8.5 17.25, off 0.5
day 2012-08-24 8.75 17.75, off 0.75
day 2012-08-27 8.5 17.25, off 0.5
day 2012-08-28 8.75 17.75, off 0.75
day 2012-08-29 8 18, off 0.5, off 0.25
leave-days 2012-08-30 2012-09-04 # hiking
day 2012-09-05 8.75 17.75, off 0.75
```

```
day 2012-09-06 8 18, off 0.5, off 0.25
day 2012-09-07 8.5 17.25, off 0.5
day 2012-09-10 8.75 17.75, off 0.75
day 2012-09-11, sick 8
day 2012-09-12, sick 8
day 2012-09-13 8.5 17.25, off 0.5
day 2012-09-14 8 18, off 0.5, off 0.25
day 2012-09-17 8 18, off 0.5, off 0.25
day 2012-09-18 8.5 17.25, off 0.5
```

Use option `-f week` to get an overview of weekly work time balances.

```
> timeflies.pl -t -f week work-log.fly
Time at work overview (week):
   week 2012-34:   16.50 worked, ----.-- leave, ----.-- sick
   week 2012-35:   25.75 worked,   16.00 leave, ----.-- sick
   week 2012-36:   25.75 worked,   16.00 leave, ----.-- sick
   week 2012-37:   25.75 worked, ----.-- leave,   16.00 sick
   week 2012-38:   17.50 worked, ----.-- leave, ----.-- sick
          total:  111.25 worked,   32.00 leave,   16.00 sick
```

Or have both weekly and monthly balances shown.

```
> timeflies.pl -t -f week,month work-log.fly
Time at work overview (week, month):
   week 2012-34:   16.50 worked, ----.-- leave, ----.-- sick
   week 2012-35:   25.75 worked,   16.00 leave, ----.-- sick
  month 2012-08:   42.25 worked,   16.00 leave, ----.-- sick
   week 2012-36:   25.75 worked,   16.00 leave, ----.-- sick
   week 2012-37:   25.75 worked, ----.-- leave,   16.00 sick
   week 2012-38:   17.50 worked, ----.-- leave, ----.-- sick
  month 2012-09:   69.00 worked,   16.00 leave,   16.00 sick
          total:  111.25 worked,   32.00 leave,   16.00 sick
```

Maybe you only want to look at one month?

```
> timeflies.pl -t -f 2012-09 work-log.fly
Time at work overview (2012-09):
2012-09-03, Mon: ----.-- worked,    8.00 leave, ----.-- sick
2012-09-04, Tue: ----.-- worked,    8.00 leave, ----.-- sick
2012-09-05, Wed:    8.25 worked, ----.-- leave, ----.-- sick
2012-09-06, Thu:    9.25 worked, ----.-- leave, ----.-- sick
2012-09-07, Fri:    8.25 worked, ----.-- leave, ----.-- sick
   week 2012-36:   25.75 worked,   16.00 leave, ----.-- sick
2012-09-10, Mon:    8.25 worked, ----.-- leave, ----.-- sick
2012-09-11, Tue: ----.-- worked, ----.-- leave,    8.00 sick
2012-09-12, Wed: ----.-- worked, ----.-- leave,    8.00 sick
2012-09-13, Thu:    8.25 worked, ----.-- leave, ----.-- sick
```

```
2012-09-14, Fri:    9.25 worked, ----.-- leave, ----.-- sick
   week 2012-37:   25.75 worked, ----.-- leave,   16.00 sick
2012-09-17, Mon:    9.25 worked, ----.-- leave, ----.-- sick
2012-09-18, Tue:    8.25 worked, ----.-- leave, ----.-- sick
   week 2012-38:   17.50 worked, ----.-- leave, ----.-- sick
  month 2012-09:   69.00 worked,   16.00 leave,   16.00 sick
          total:   69.00 worked,   16.00 leave,   16.00 sick
```

You only need weekly totals in that one month?

```
> timeflies.pl -t -f week,2012-09 work-log.fly
Time at work overview (week, 2012-09):
   week 2012-36:   25.75 worked,   16.00 leave, ----.-- sick
   week 2012-37:   25.75 worked, ----.-- leave,   16.00 sick
   week 2012-38:   17.50 worked, ----.-- leave, ----.-- sick
          total:   69.00 worked,   16.00 leave,   16.00 sick
```

2.4 Logging Activities

Log comments are a good way to keep track of things you don't want to forget and have accessible and also aligned with your work time line. Log comments have no work effort assigned to them, though. So you cannot use them in any way for calculations of effort spent.

You use *work packages* and *activities* to connect the time you work with the work packages you work on: first, you define your work packages, then you use *activity* lines in the day blocks instead of log comment lines.

A work package definition is a line starting with the keyword `work-package` (or its abbreviation `wp`) followed by a work package name.

An activity line starts with a single dash character `-` followed by one or more spaces. This is followed by a work package id and a duration. This is optionally followed by a semicolon and some activity comment.

See below the converted example work log file.

```
wp regression-tests
wp meetings
wp quality-task-force
wp problem-reports
wp development
wp other

day 2012-09-19 8.5 17.25
- regression-tests 4; updated
off 0.5
- other 3.5; fixed build scripts

day 2012-09-18 8.75 17.75
```

```
- problem-reports 2; wrote unit test to reproduce problem report 2012-0098
- problem-reports 2.5; fixed problem report 2012-0098
off 0.75
- development 3; added HTML output option to object dumper
- other 1; discussed implications of Java 1.7 rollout

day 2012-09-17 8 18
- meetings 2.0; weekly team meeting
off 0.5
- quality-task-force 6
off 0.25
- other 1.25; code review: server side includes
```

Option `-w` tells *TimeFlies* to calculate the times you have been working on the different work packages:

```
> timeflies.pl -w work-log.fly
Work package summary (all):
  25.25 : ALL
        4.00 : regression-tests
        2.00 : meetings
        6.00 : quality-task-force
        4.50 : problem-reports
        3.00 : development
        5.75 : other
```

To also show the activities contributing to the different work packages, use option `-a`:

```
> timeflies.pl -w -a work-log.fly
Work package summary (all):
  25.25 : ALL
        4.00 : regression-tests
               - 2012-09-19 4.0; updated
        2.00 : meetings
               - 2012-09-17 2.0; weekly team meeting
        6.00 : quality-task-force
               - 2012-09-17 6.0
        4.50 : problem-reports
               - 2012-09-18 2.0; wrote unit test to reproduce problem report 2012-0098
               - 2012-09-18 2.5; fixed problem report 2012-0098
        3.00 : development
               - 2012-09-18 3.0; added HTML output option to object dumper
        5.75 : other
               - 2012-09-19 3.5; fixed build scripts
               - 2012-09-18 1.0; discussed implications of Java 1.7 rollout
               - 2012-09-17 1.25; code review: server side includes
```

To check whether you have allocated all your working time to work packages, use option `-c`:

```
> timeflies.pl -c work-log.fly
Day check (all):
*** 2012-09-18, Tue : worked =  8.25, tasked =  8.50, delta =  0.25
*** 2012-09-19, Wed : worked =  8.25, tasked =  7.50, delta = -0.75
```

This shows that on two days the time at work and the time worked on work packages are differing.

2.5 Work Package Breakdown

In the previous section, work packages have been defined as simple, atomic, named items. A work package can be subdivided and refined hierarchically. See the following example.

```
wp md; MightyDigester: digests inputs of all sorts
    in; read supported input formats
        xml
        json
        dottxt; dotted text format
        binary
    proc; processing modules
        stats; processing statistics
        phase-1; rough break-down
        phase-2; particle recombination
        phase-3; regrouping and amalgamation
    out; write supported output formats
        xml
        json
        text
        binary
    mmi
        gui
        cmdline
```

The items in this work package hierarchy can be referred to in activity lines as dot-delimited work package path names.

Following, a piece of work log for the above project.

```
day 2012-07-01 8 17, off 1
- md.in.xml 4; updated to new XSD
- md.out.xml 3; updated to new XSD
- md.mmi.cmdline 1; XML options
day 2012-07-02 8 17, off 1
- md.in.json 5; first minimal implementation
```

```
- md.proc.stats 1.5; line counting
- md.mmi.cmdline 1.5; statistics options
day 2012-07-03 8 17, off 1
- md.in.xml 4; adapted includes
- md.out.xml 3; normalised host node structure
- md.out.text 0.5; don't use TAB any more
- md.mmi.cmdline 0.5; text and xml options
```

Assume file `prj-mighty-digester.fly` contains the work package defintions and the work log itself is kept in `work-log.fly`. The work package summary can be calculated with option `-w` (which was also used in the previous example).

```
> timeflies.pl -w prj-mighty-digester.fly work-log.fly
Work package summary (all):
  24.00 : ALL
      24.00 : md; MightyDigester: digests inputs of all sorts
          13.00 : in; read supported input formats
               8.00 : xml
               5.00 : json
           1.50 : proc; processing modules
               1.50 : stats; processing statistics
           6.50 : out; write supported output formats
               6.00 : xml
               0.50 : text
           3.00 : mmi
               3.00 : cmdline
```

And here the same with activities shown.

```
> timeflies.pl -w -a prj-mighty-digester.fly work-log.fly
Work package summary (all):
  24.00 : ALL
      24.00 : md; MightyDigester: digests inputs of all sorts
          13.00 : in; read supported input formats
               8.00 : xml
                      - 2012-07-01 4.0; updated to new XSD
                      - 2012-07-03 4.0; adapted includes
               5.00 : json
                      - 2012-07-02 5.0; first minimal implementation
           1.50 : proc; processing modules
               1.50 : stats; processing statistics
                      - 2012-07-02 1.5; line counting
           6.50 : out; write supported output formats
               6.00 : xml
                      - 2012-07-01 3.0; updated to new XSD
                      - 2012-07-03 3.0; normalised host node structure
               0.50 : text
                      - 2012-07-03 0.5; don't use TAB any more
```

```
            3.00 : mmi
                3.00 : cmdline
                       - 2012-07-01 1.0; XML options
                       - 2012-07-02 1.5; statistics options
                       - 2012-07-03 0.5; text and xml options
```

3 Reference

3.1 Command Line Options

Time filters
- Summary filters
- User filters
- Show work packages
- calculate work packages
- check days
- tally days
- indentation

3.2 File Syntax

wp, work-package
- day, off, leave, sick
- block-leave
- log comment: ;
- activity: -