

# Investors' report

David Kastrup

August & September 2013

My plans for catching up with my delayed reports has not really worked out because of lots of work needing to get done, so I have now decided to put out two combined reports after all. If you use the arrival of a report as your reminder for scheduling a payment, try factoring the double report into your plans. Like with the previous reports, the largest sum in these reports is actually from a one-time payment distributed across more than a year.

## August

<u>Fixed payments (€)</u>	
300	
2×200	
100	
2×50	
44	
3×25	
2×20	
<u>3×10 7</u>	
1096	
	<u>Variable plans (€)</u>
	<u>25 + 0×50 (target €1200 exceeded in July)</u>
	25

Totals €1121

## September

Fixed payments (€)	
<hr/>	
300	
2×200	
2×150	
2×100	
75	
3×25	
24	
2×20	
4×10	
<hr/>	
1454	
	Variable plans (€)
	<hr/>
	25 + 1×50 (target €1200 missed in August)
	<hr/>
	75

Totals €1529

So the payments in September have managed to compensate for what has been missing in August for making ends meet. I have not yet looked at the numbers for October, though, and I hope that my somewhat sluggish update schedule (and the resulting double update here) will not cause too much of a hit.

At any rate, at this point of time there is no question that I'll keep overseeing the release of 2.18.0 to its completion. It's not like there isn't a lot of work after that, but we are in a good position to see releases of 2.18 appear in the spring updates of GNU/Linux distributions.

## Development results

There has actually been a lot going on these months. Let's take a look.

### August

Calling

```
git shortlog origin --since 2013/08/01 --until 2013/09/01 -n
```

in a checkout of the project repository lists 37 commits.

Most of those are general fixes including a tricky garbage-collection related crash, cleanups and small improvements, but there also a few noteworthy improvements.

I'll quote a few (related) items from the "Changes" list:

- A number of shorthands like (, ), |, [, ], ~, \{, \} and others can now freely be redefined like normal commands. An example would be

```
"\{" = (  
"\}" = )
```

```

"(" = \melisma
")" = \melismaEnd

\new Staff <<
  \relative c' {
    c8 \{ d e f \} % slurred
    g ( a b c ) % no slur, but with melisma
    c,1 \bar "|."
  }
  \addlyrics { Li -- ly -- pond. }
>>

```



- The articulation shorthand for `\staccatissimo` has been renamed from `-|` to `-!`.
- Tempo change ranges are now written as `\tempo 4 = 60 - 68` rather than the previous `\tempo 4 = 60 ~ 68` (the difference is the character for indicating the range).

The first item makes it much easier to experiment with syntax changes or to add additional functionality or debugging code to existing commands. The last two items were necessary consequences of implementing the first item but make good sense anyway.

Another usability change can probably best be illustrated by quoting its commit message:

```

commit 95e4b7f9d22b54ae991ac6d1a8026667eadc1e8f
Author: David Kastrup <dak@gnu.org>
Date: Sun Aug 25 19:36:11 2013 +0200

```

```

Issue 3517: Parse composite music in context modifications in \notemode

```

```

This allows for context mods like

```

```

\new Staff \with { \transposition f' } { c'4 d' e' f' }

```

It was previously possible (in 2.16 I think) to use simple music expressions like `\voiceOne` or `\accidentalStyle modern` inside of context modifications, but you could not make use of actual note names.

## September

Calling

```

git shortlog origin --since 2013/09/01 --until 2013/10/01 -n

```

in a checkout of the project repository lists 44 commits: a rather busy month.

The “context modifications in `\notemode`” change was augmented with

```
commit 3b617fc949192eacf627413dac0420ebc54aebf6
Author: David Kastrup <dak@gnu.org>
Date: Sun Sep 15 15:53:51 2013 +0200
```

Issue 3555: Parse context definitions and context modifications in `\notemode`

Since context modifications and context definitions are somewhat more concerned with musical content, it makes sense to parse them in a mode where note names are recognized and can be used in music and scheme function arguments.

The motivation for this was to be able to write

```
\new TabStaff \with {
  stringTunings = \stringTuning <c' g' d'' a''>
} ...
```

which previously required writing

```
\new TabStaff \with {
  stringTunings = \stringTuning \notemode <c' g' d'' a''>
} ...
```

The change in August was not helpful for that purpose since `\stringTuning` is not actually a music expression but a Scheme function returning a list of pitches.

There were also a surprising number of documentation fixes for arriving at a status suitable for a stable release, some streamlining of the parser and its behavior, a number of smaller and more involved bug fixes and changes in behavior, and some work on making the Scheme function `make-music` quite more convenient for some common programming tasks.

## Ongoing tasks

LilyPond 2.18 is, of course, still on the agenda. I branched off the release branch from our main development recently, so it’s not unreasonable to expect seeing an actual release next month. As you can see from the reactions to this announcement, the long leadup to the release is grating on everybody’s nerves, but we still had a number of outstanding issues that needed addressing. We are not actually through with all of them, but it would appear that we have leads on pretty much everything that’s really important.

The stock “Metapost” program from the `TEXlive` 2013 release creates quite deficient fonts for LilyPond (running the `tlmgr` update manager can fix this for custom installations of `TEXlive`). It turns out that exactly this version is what has been released with Ubuntu 13.10, leading to an included compilation of LilyPond 2.16.2 that comes with severely defective fonts. We are currently working on damage control. Thanks to catching the ear of a responsive Ubuntu developer, fixed versions should appear in the

distribution updates soon. On the plus side, the long-broken PDF previewer Xpdf has been released in a working state this time. A backport of its fix to 13.04 has been created as well, but has not yet made it through the pipelines.

## Perspectives

Well, there are a few changes and tasks piling up for the time after releasing 2.18. It is actually quite embarrassing that we have no version of LilyPond working with GUILE 2.0 (which has been available for several years now) yet: we are still firmly entrenched in working with GUILE 1.8 and are basically the main reason GNU/Linux distributions still distribute it. There also have been discussions on our mailing lists recently whether it would make sense to base our issue tracking and review processes on other tools and platforms. After the release of 2.18, there should be more opportunity for experiments in that area.

Since we experienced a fair bout of tension with the preparations for 2.18, it might also be good to rethink out development processes. It is not clear to me, however, that we can do much better with the limited developer power we have available. At the current point of time, there is no similarly large project in the queue for 2.19 like the skyline positioning methods mostly written by Mike Solomon and finally committed to LilyPond 2.17.1 were.

That makes it somewhat more likely that the next stable release cycle will be able to shift some focus to further stabilization and projects like the GUILE 2 migration, possibly also to improvements of tools and workflows.

Two already impressively active projects are the Denemo music notation editor employing LilyPond as its backend and Frescobaldi, “a LilyPond sheet music text editor”, both available as Free Software.

If you are interested in their respective area of work and have not looked at them for a long time, it might be worth revisiting their current offerings.

## Thanks

as always for making it possible for me to continue my work. I hope that we’ll soon be seeing the 2.18 milestone as a result of your dedication.

David Kastrup