Infrastructure - Task #3512

Weblate translation entry textbox doesn't properly support Hebrew

2021-03-13 21:32 - Eyal Rozenberg

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Christian Lohmaier	**% Done:**	0%
Category:	Weblate		
Target version:	Q3/2021		
Tags:		**URL:**	

Description

When you enter Hebrew translation text (and, in fact, when you look at the glossary entries) which have Hebrew punctuation marks such as MAQAF - they are not rendered properly. This is likely due to the use of a font which lacks the appropriate glyphs.

History

#1 - 2021-03-16 15:01 - Sophie Gautier

This seems corrected by this issue https://github.com/WeblateOrg/weblate/issues/4772
Could you comment on the bug if this is not the case? Thanks in advance

#2 - 2021-03-16 15:40 - Eyal Rozenberg

Sophie Gautier wrote:

> This seems corrected by this issue https://github.com/WeblateOrg/weblate/issues/4772

That issue is about Hebrew quotation marks. Still, I've made a comment about MAQAF there, since Yaron mentioned it in a comment of his own.

#3 - 2021-03-16 16:48 - Sophie Gautier

Eyal Rozenberg wrote:

> Sophie Gautier wrote:
>
>> This seems corrected by this issue https://github.com/WeblateOrg/weblate/issues/4772
>
> That issue is about Hebrew quotation marks. Still, I've made a comment about MAQAF there, since Yaron mentioned it in a comment of his own.

Thanks! And I see that Adolfo recommended you to open another issue and gave you a workaround until it's corrected.

#4 - 2021-05-31 11:25 - Christian Lohmaier

- Assignee set to Christian Lohmaier

Is this still a problem after the latest updates?
And if it is a font problem: what (free) webfont would be suitable to use instead?

(also please provide a link to a sample string, and ideally also a png of how it should be rendered, since honestly I wouldn't able to tell bad from good rendering)

#5 - 2021-05-31 19:55 - Eyal Rozenberg

Christian Lohmaier wrote in #note-4:

> Is this still a problem after the latest updates?

Yes, it's still a problem. I just checked again. See The glossary or just a certain individual strings

#6 - 2021-06-01 15:19 - Christian Lohmaier

- File Bildschirmfoto von 2021-06-01 15-08-31.png added

You unfortunately didn't provide a "how it should be" rendered, even when using a "Hebrew" font, it pretty much looks the same to me, the vertical alignment is a little different, and I can only assume that that is the problem here...

Anyhow https://jsfiddle.net/04bhy6pL/ is a short test as to see whether the rendering is OK and what kind you prefer. (western characters look a little too heavy when using just the Rubik font, but then again I have no idea how mixed text typically looks like for Hebrew) - and I just picked that font because it was first on the list when I limited the fonts to language Hebrew, sans-serif…
Not sure whether it is worth fixing, since after all I'd assume the users' browser would have a default set of fonts (the "sans-serif" alias) that would render their language properly?

#7 - 2021-06-01 15:59 - Adolfo Jayme Barrientos

This Source-derived design has Hebrew characters. Maybe we could add it to our Weblate while (if at all) upstream considers it. WDYT, Eyal, Cloph?

#8 - 2021-06-02 14:19 - Christian Lohmaier

Thanks for the pointer, Assistant font of course would be more obvious choice if it is indeed meant as an addition to the Source Sans Pro font - updated jsfiddle with the font as additional example: https://jsfiddle.net/pbdja8xf/

For some reason though the font-weight is off compared to Source Sans, weird for a font that is meant to be used as an extension of an existing one. (just add two paragraphs with latin text, one set to font-family: "Source Sans Pro" and the other to font-family: "Assistant" - could also be that the css handed out by the google URL uses the wrong fonts, but both are specified in the returned css as style normal, with a weight of 400, but still pretty different in browser.

As to upstream: Since the font doesn't include Hebrew characters at all, and it is possible to just limit Assistant to Hebrew subset it might be worth adding it to upstream, although that then rises the question: when to stop adding fonts. Easy to argue for Assistant "it is an extension to the font you're already using", but of course other for other code-ranges it won't be that straight forward..
That brings me back to my other initial question: can't we "assume the users' browser would have a default set of fonts (the "sans-serif" alias) that would render their language properly"? Guess the answer is "no"?

#9 - 2021-06-02 16:46 - Eyal Rozenberg

Adolfo Jayme Barrientos wrote in #note-7:

> This Source-derived design has Hebrew characters. Maybe we could add it to our Weblate while (if at all) upstream considers it. WDYT, Eyal, Cloph?

The MAQAF looks fine. Cantillation marks are missing, but I don't think we need those anywhere in the UI localization.

And - unfortunately, it seems that for now, users' browsers' default fonts can't be assumed to render Hebrew correctly. They can be assumed to render it legibly, though. So it is legitimate to say: "This is a browser problem, not a weblate problem". And I'm not even entirely against that, but then - I would like that someone higher-up in the project make a proper complaint to Mozilla, Google, Microsoft and Opera about this issue (after verifying that the problem is observed with all of their browsers.

#10 - 2021-06-23 11:49 - Florian Effenberger

- Status changed from New to Feedback

- Target version set to Q3/2021

Cloph mailed Sophie for some feedback

#11 - 2021-07-19 12:04 - Florian Effenberger

- Status changed from Feedback to In Progress

Will be done after 7.2

#12 - 2021-09-13 07:16 - Adolfo Jayme Barrientos

The newly installed Weblate now uses a locally-served Source Sans 3 (instead of the old “Source Sans Pro” that is currently served on e.g. Google Fonts). Any comparison with the Assistant font should now be done with “Source Sans 3”, but still I don’t think anybody should expect a perfect down-to-the-pixel match between the two families — that is unreasonable.

#13 - 2021-10-15 19:41 - Eyal Rozenberg

> The newly installed Weblate now uses a locally-served Source Sans 3

The issue still persists.

#14 - 2021-10-18 11:09 - Christian Lohmaier

- Status changed from In Progress to Resolved

Hebrew now uses the Assistant with the Source font as fallback

#15 - 2021-10-18 11:33 - Eyal Rozenberg

Christian Lohmaier wrote in #note-14:

> Hebrew now uses the Assistant with the Source font as fallback

I don't know what that means, but the bug as described still persists. Please explain or unresolve it.

#16 - 2021-10-18 11:42 - Christian Lohmaier

- File Bildschirmfoto von 2021-10-18 11-35-24.png added

- File Bildschirmfoto von 2021-10-18 11-34-39.png added

Weblate uses the Assistant font for the translation box as well as the nearby string listing. You did signal that the Assistant font renders the characters OK, so that's why I did use it.

The font is only added with the Hebrew subset, so latin text in hebrew string would still use the Source Sans Pro font, that's meant with the fallback.

#17 - 2021-10-21 22:09 - Eyal Rozenberg

Christian Lohmaier wrote in #note-16:

> You did signal that the Assistant font renders the characters OK, so that's why I did use it.

I haven't actually checked what font is being used in the textboxes. But - your screenshots doesn't show any Hebrew punctuation marks at all - neither correctly nor incorrectly rendered.

But, wait, now I do see the MAQAF rendered correctly! Not when I select text, and the cursor position is messed up, but still. So, no need to unresolve I guess.

#18 - 2021-10-28 16:15 - Christian Lohmaier

can you check again? the textedit mode wasn't included in the css rule, so that used the old fallbacks, and depending on your system that might have different metrics and cause some offset in cursor placement.
If you still have the problem, please provide a short screen recording demonstrating the problem.

#19 - 2021-10-28 23:15 - Eyal Rozenberg

Christian Lohmaier wrote in #note-18:

> can you check again? the textedit mode wasn't included in the css rule, so that used the old fallbacks, and depending on your system that might have different metrics and cause some offset in cursor placement.
> If you still have the problem, please provide a short screen recording demonstrating the problem.

Seems fine now, also when selecting text. Thank you.

#20 - 2021-10-30 23:15 - Christian Lohmaier

- Status changed from Resolved to Closed

thx for checking → closing.

Files

shot2.png	31.4 KB	2021-03-13	Eyal Rozenberg
Bildschirmfoto von 2021-06-01 15-08-31.png	430 KB	2021-06-01	Christian Lohmaier
Bildschirmfoto von 2021-10-18 11-35-24.png	209 KB	2021-10-18	Christian Lohmaier
Bildschirmfoto von 2021-10-18 11-34-39.png	246 KB	2021-10-18	Christian Lohmaier