

Advancing the Advanced...

DECEMBER 19, 2017 ~ PAULFILKIN



Some time ago the SDL AppStore team created an opensource site where they make the source code available for virtually all the apps they create for the SDL AppStore. You can find the site here, <https://sdl.github.io/Sdl-Community/>, along with links to the apps themselves and also the sourcecode which can be pulled by any developer so they can make their own enhancements and improvements based on a good headstart. I love this concept, but have to say I'm a little disappointed by the lack of active participation from other developers in pushing their own work back into the apps to share the improvements. At least I'm disappointed in general, but there are exceptions even if they have been carried out by the AppStore team themselves! The best exception and example of what can be achieved is around the Advanced Display Filter that can be found in Studio 2017.

The Advanced Display Filter wasn't actually an app in the first place, it was always part of the core product. But when it was developed the potential for lots of other improvements was obvious and the original developer had to really constrain himself to stop adding features every time someone thought of something else we could add! So we decided to opensource the code so that external developers could extend it as they saw fit. In the next months I actually haven't seen any improvements to the source code, although I have seen a few strange attempts to use the capability of the filter in other ways with separate apps for different types of filters. None of them were put into the appstore so in the end we gathered up all the ideas we had seen people asking for and extended the app ourselves... and of course we pushed the **source code** back into the SDL github site for any developer to use as they see fit. The new features are really cool so I'll go through them one at a time

and you can judge for yourselves. If you just want to dive right in and have a play then you'll find the plugin on the [SDL AppStore](#)... enjoy!

Reverse Filter

The first thing I want to mention is the “Reverse Filter” as this is something I have seen coming up for years, starting with the original basic filter and even the new Advanced Display Filter in Studio 2017. The basic idea is that if you filter on something, for example all segments containing numbers, but you actually want all segments that don't contain numbers then it's often a lot harder to achieve as the regular expressions for a negative result can be more complex to write. So if you could filter on numbers as this is easy but then flip a switch to show the opposite then the problem would be solved! Most of the requests I have seen for this relate to the regular expression search tab, but we thought it would be even better if it applied to any filter that was used. So we put it here:

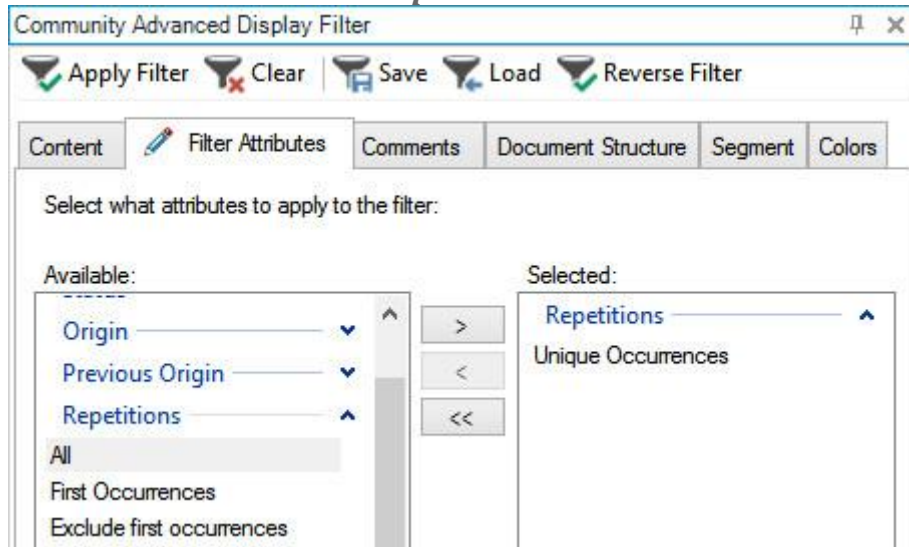


This way it doesn't matter which of the filters are applied, you just use the filter settings you need and then either click **Apply Filter** and then **Reverse Filter** or just click **Reverse Filter**. I mentioned this enhancement first because every feature of the existing Advanced Display Filter (which are also part of this new version) and every new feature I mention below can be reversed with this action. Very smart, very obvious... love this enhancement alone!

Filter Attributes

This is an existing tab with the ability to filter on all kinds of statuses, translation origin, review features, repetitions, locked content etc. It was missing an option in the **Repetitions** group to be able to find all Unique

Occurrences rather than just *All* repetitions or *First Occurrences* or *Exclude First Occurrences*. It was missing the ability to filter and show the first occurrence of every segment that occurred even if that occurrence was one. So we added it... *Unique Occurrences*:



To explain this a little better assume you have the following segments:

- #1 – This is a test
- #2 – This is a test
- #3 – They were tests
- #4 – This is a test
- #5 – You just said that

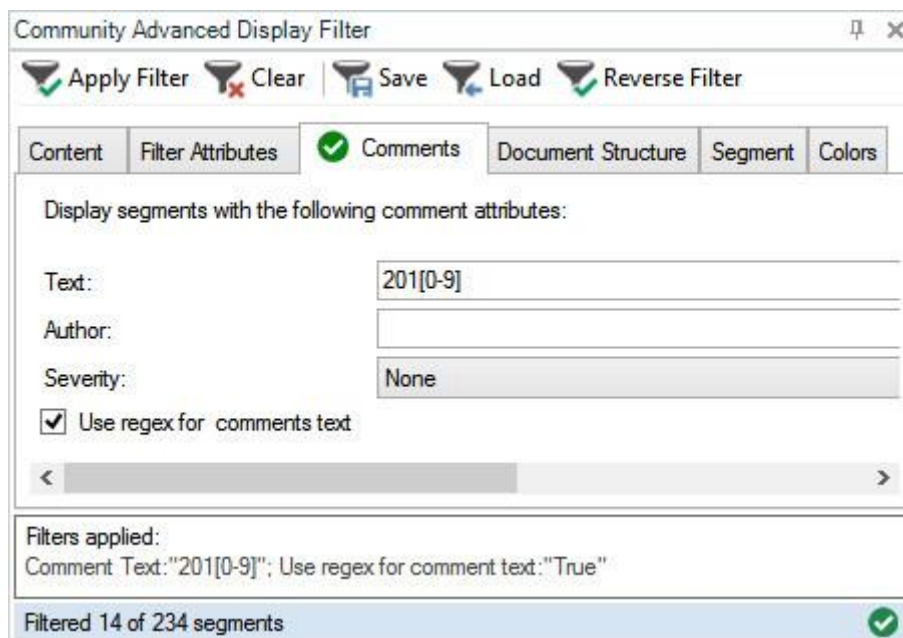
If you apply this new filter on the text you would get this:

- #1 – This is a test
- #3 – They were tests
- #5 – You just said that

So you get all segments but only the first repetition of each repetition. This feature was thought to be really useful when working with QA tools such as the [Antidote Verifier](#) so you don't have to deal with the same errors again. You can allow Studio to deal with the correction afterwards.

Regex for Comments text

Another existing tab, the Comments tab. This allows you to filter on segments with comments but can be based on specific text, author or severity. The enhancement here was allow you to search for the text in the comments using regular expressions. For example, if I had a document full of comments containing references to dates between 2010 and 2019, then I could use a filter to find only the segments with these particular comments by searching for only these dates in the comments as opposed to all the comments and then having to inspect them, or search for each year one at a time:



And of course clicking on **Reverse Filter** would find all the segments without these particular comments.

Segment actions

The new tab called "**Segment**" offers a variety of search features that relate to the segments in the SDLXLIFF. They're in one tab as the feedback we had during testing was that they are all related and would be better in one tab to save clicking around through too many tabs... you be the judge if this makes sense or not... I prefer them in one place too. I'll just take them one at a time:

Number Options

These relate to the segment numbers in the left hand column. You can filter on odd numbers, even numbers or groups of numbers so this makes it possible to filter on any random set of segments you like:

Number options

Even segments

Odd segments

Grouped lists

Example of grouped list: 1-2, 17-45, 56, 100-150

1-15, 20-25, 115-232, 250, 300, 500

In the screenshot I showed the Grouped Lists because you need to make sure that you check the box **Grouped lists** before the box is active and you can write the segments you want to filter on in there.

Segments Option

This group offers several useful options for filtering on segments that have been altered from the original by splitting or merging them and also for identifying segments where the source text equals target text:

Segments option

Split segments

Merged segments

Merged segments including paragraph merge

Source text equals target text

Is case sensitive

Could be useful for a Project Manager getting SDLXLIFF files back where this action was not permitted, especially if it was merging across paragraph breaks as it would be very tedious to find them in a large file. The **Merged segments** will find segments that have been merged within a paragraph. The **Merged segments including paragraph merge** will find

segments merged across paragraphs breaks in addition to normally merged segments. The reason for this difference is that the paragraph break merge is a little different to a normal merge and is really just an automation of a manual workaround where the segments are not actually merged at all. What happens is that the text in the second segment selected is cut leaving a completely empty segment which is then locked and hidden, then the text is pasted into the first. You can read more on this if you're interested in [this article](#).

Source text equals target text

This feature has been grouped together with the *Segments option* and I think it's quite obvious and probably very useful if you come across a situation where you need this!

Source text equals target text

Is case sensitive

The *Source text equals target text* can filter on segments where the source and target text (including tags) are the same. You may well ask why did we do this as there is a *Copied from source* Filter Attribute already? Well there is a difference... the latter filters on segments where the Origin attribute on the segment is source copied to target. This new feature looks at the content, not the origin, and filters on segments where the source is equivalent to the target and this potentially makes it quite a powerful feature for QA, particularly where the SDLXLIFF files were translated in other tools.

Is case sensitive provides the ability to only filter where source equals target in more than the content. So if you were looking for lowercase source to become uppercase target then you could use this to filter on all the segments where the source segments were copied to target without changing the case.

Please add fuzzy range

The Filter Attributes tab offers many options for filtering on the Origin of the segment based on match values. You can, as part of this, filter on Fuzzy Matches but you can't filter on a specific fuzzy match or range of fuzzy matches. So this filter allows you to do just that:

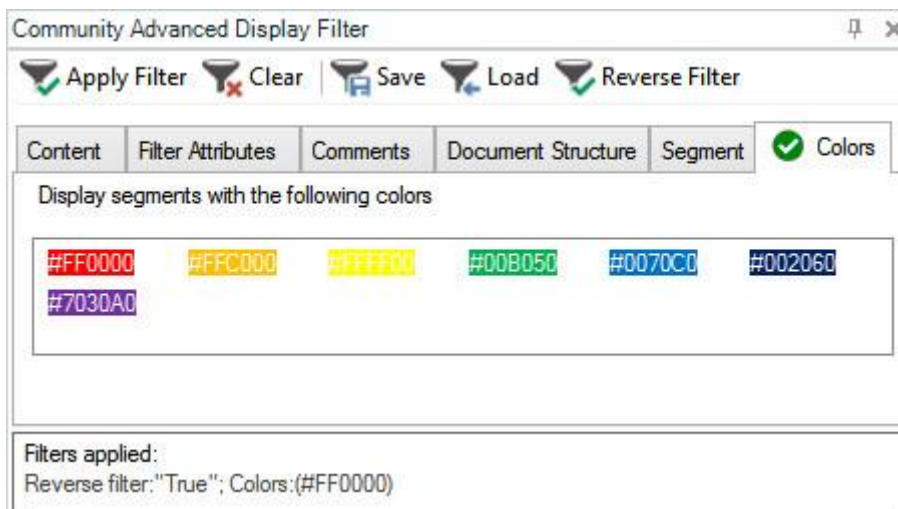
Please add fuzzy range

Fuzzy values between and

If you wanted to find all the 99% matches in your SDLXLIFF then you would put 99 into each of the boxes and apply the filter. If you wanted to find all the fuzzy matches between 50% and 75% then you would put 50 in the left box and 75 in the right and apply filter. Very straightforward and logical.

Colors

The final tab, and probably the most problematic to implement so that it worked for all filetypes, is the ability to filter segments based on the colour of the text within the segment. The whole segment does not need to be coloured, it's enough for a single letter to be that colour and the filter will pick it up and display that segment:



Andrea-Melinda (the developer) experimented with colour pickers, colour wheels, palettes and all sorts before we decided the best approach was simply to look in the file, see what coloured text was in there and then display the colours found so they can be selected. You can select multiple colours if you like and when you apply the filter you should only see segments containing the coloured text of your choice... and of course you can use the **Reverse Filter** here too!

It all seemed relatively straightforward once we decided on the right approach and then we tested some different file formats... back to the drawing board! Quite a tricky exercise in fact and I hope the source code will be useful for any developer looking to implement a solution based on the colour of the text!

Summary

So that's it... 8 new features added to the display filter (and a few bug fixes to the original version in Studio 2017). All that's left is to give you a tip on installing. The app is an **sdlplugin** so you double click it to install into Studio, but it won't overwrite the existing Advanced Display Filter. This was not possible to do, and we could not disable the existing one either, so when you install it you will have two advanced display filters:

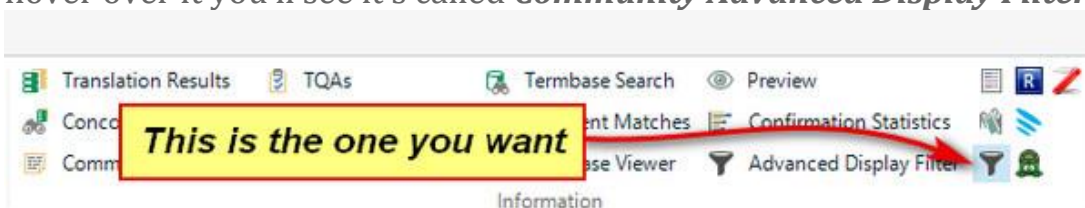
1. the out of the box **Advanced Display Filter**
2. the new **Community Advanced Display Filter**

The new one extends the existing so you don't need them both. The easiest solution is to close the out of the box window so that it's not visible in the user interface anymore:



Click here to close it

Then activate the new one by clicking here on the same icon but when you hover over it you'll see it's called **Community Advanced Display Filter**:



Then position the window wherever you want it. If I was critical of anything it would be that I wish there was a better way to do this so it either deactivated the original automatically, or overwrite it, but there isn't. This may change in the future but for now you have a little manual effort... but it's worth it!

I hope developers also see these changes as useful and I'm keeping my fingers crossed that we'll see more enhancements to the source code we make available so that these plugins can be updated and of benefit to everyone.