For autocomplete suggestions to work well, they must help the user narrow down the available options from a dataset to a manageable list of options. This means that to work effectively, it must be smart, it’s use must not be frustrating, its use must be intuitive. This means we’d like to use a few things which we do not see how to do with the Syncfusion implementation given the samples (including use within sfTextBoxExt due to the deprecation of AutoComplete as a separate control. A few things we are trying to understand if currently supported.

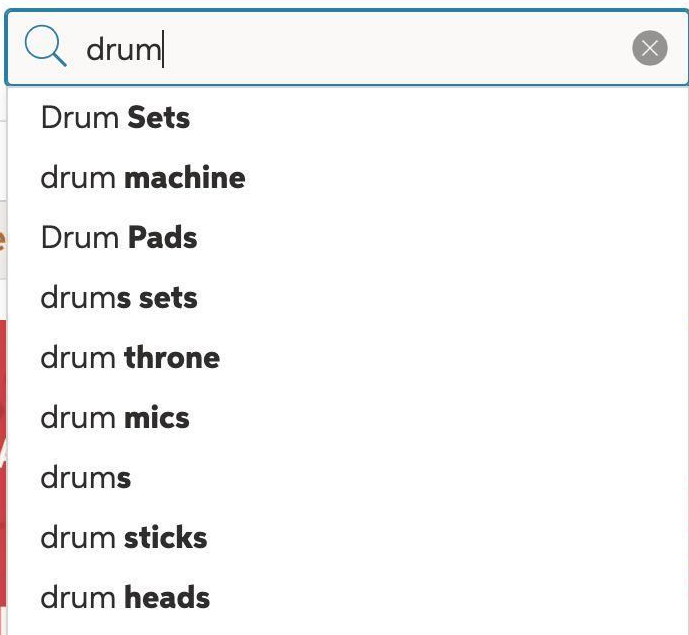
1. Keep the List Manageable

Because the desktop autocomplete feature should expand to fit its options (and we prefer not to use of a scrollbar), the number of options should be limited so they don’t overflow the user’s viewport.

We would like to be able to limit the number of choices displayed to a number, ie 10 items.

1. Style Auxiliary Data Differently

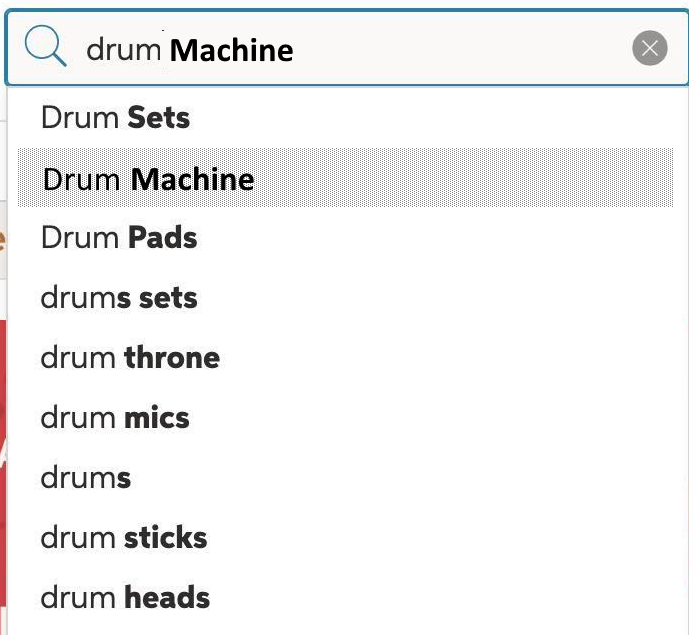
Styling helps provide subtle cues that orient users to the different types of suggestions and data shown in autocomplete. Any alternate or auxiliary information, such as scope suggestions (i.e., “search within [XY] category”) or the number of matches, should be **given a unique style** so the user can easily determine what is (and isn’t) part of the actual search query suggestion. Ie:



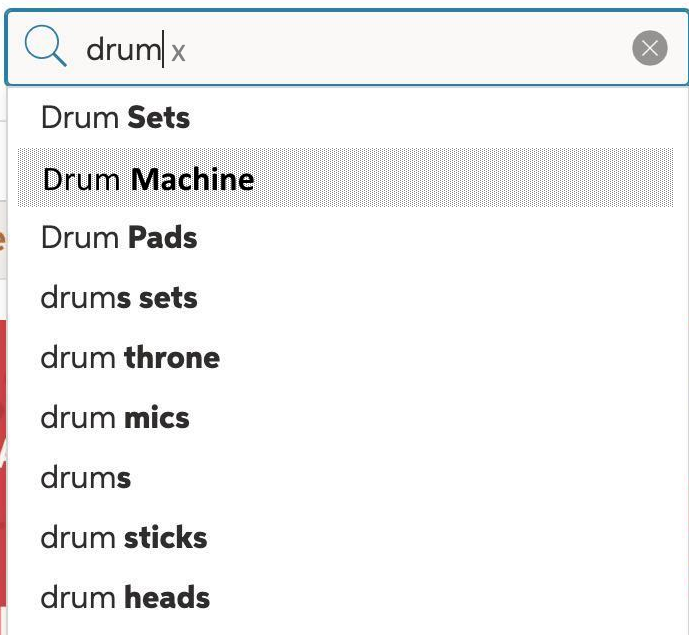
1. Indication of active selection so it can be taken without forcing the user to navigate to get to it and select.

Clearly **indicating the active suggestion is crucial** — it provides the user with visual feedback and lets them know what is active, whether they’re hovering over suggestions with a mouse or using the keyboard to navigate the list.

Currently the Syncfusion control fails at this miserably. If you don’t navigate to a selection, you get what you typed – its not intuitive, not what is expected, It wastes user time being so inconsistent with other implementations on the desktop, web and mobile devices. It frustrates users. Either we need to highlight what is currently selected within the list or copy the info (split styled as discussed above to the text area so on enter or exiting the field, it will be selected.

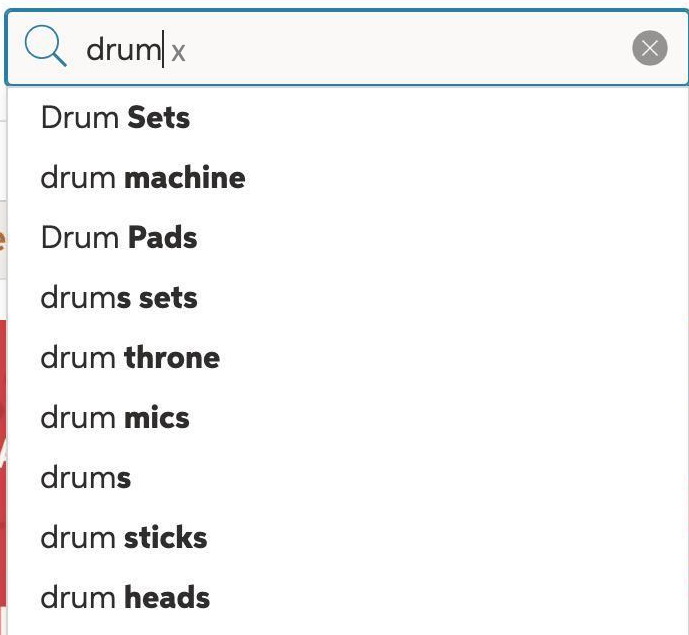


Or in the case where text starts to diverge from exact match of what is already typed, it’s just the highlight bar that was highlighted before the text entered diverged from a complete match:



1. An option to tell the control that a character(s) that don’t match does not clear the available choices

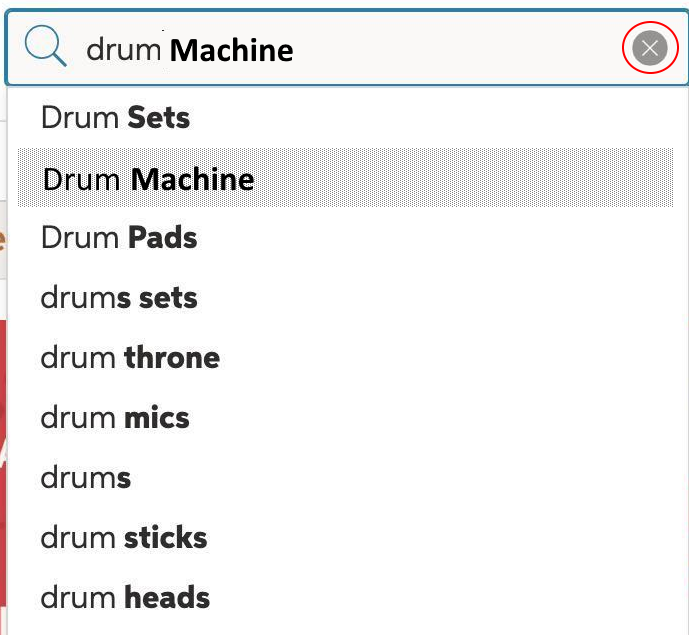
There needs to be an option to still show the choices based on what does match. In this case a user may need to navigate to a selection since they have only partially matched, but that is fine. It still makes the use of the control more intuitive.



Note the lighter color of the letter(s) typed that don’t match anything – this is in addition to the split styling already mentioned.

1. Enable Easy Removal of Queries *of the search text.*

Some users will change their search strategy mid-query, wishing to **cancel out of search** and get autocomplete suggestions out of their way. When users fully have to back up character by character or use the mouse to highlight a portion of the text to delete query terms, it’s a bunch of steps which should be simpler. We n*ote that this becomes a little less critical when #4 is working since you don’t lose available options. We currently use ESCape for this which isn’t necessarily intuitive, but an [x] at the right of the control could do this as well. (in our examples above, but highlighted below as well:*

**