L2/24-264

Text Terminal Working Group

Author: Fraser Gordon (chair)
Date: 2024-11-06

The Text Terminal Working Group (TTWG) has been meeting on a roughly monthly basis. Among regular attendees, we have a good cross section of expertise, including authors of terminal emulators, text rendering libraries and text shapers. More participants are very welcome and we operate a wide membership policy: we will invite anyone with interest and expertise in related fields to the working group, either to actively participate or observe the work in progress and offer feedback. There are ongoing discussions with the Unicode infrastructure team to ensure we have processes in place for this.

As a refresher: we are working to define algorithms for consistent and correct display of complex (shaped, bidirectional and/or “wide”) text in text terminals and other applications that display text on a fixed-width grid.

So far, our focus has been on ensuring that the problem domain is fully understood and that we are aware of how existing software handles this problem. There are a number of cases to be handled, some of which touch on areas outside the scope of Unicode and related to font or system support. For example, emoji modifier sequences and regional indicator symbols can vary widely in their presentation.

Additionally, applications have different demands from the terminal. Some simply want to write text to the terminal and have it do the right thing (e.g. `ls`) while others want sophisticated layout (TUI applications such as text editors). Both need to be accommodated.

In order to accomplish this, we will likely have to specify behaviour in areas that are unusual for a UTS. In particular, we see the need to set down rules for:

- Fonts
- Signalling between applications and terminals

In the case of fonts, we would establish a set of constraints that a font must satisfy in order to be “text terminal compatible”. These would generally relate to the font’s metrics and how it maps graphemes to glyphs so that grid layout can be performed without any information about the font (which, in general, will not be available to the application).

For signalling, a terminal needs to know if the foreground application is simple, where the terminal will be handing the layout internally, or complex, where the application takes on some of the layout responsibilities. The traditional method of accomplishing this type of control is to use in-band escape sequences.

On both of these items, feedback from the UTC is sought on appropriate ways of specifying or recommending this behaviour.

We have also looked at the properties available in the UCD and their sufficiency for performing the necessary layout. The consensus is that none of the properties (in isolation) capture what is needed; in particular, the East Asian Width (EAW) property does not meet our needs. Additional properties will be needed. These would be derived properties (EAW, for example, is correct in many straightforward cases and would be one of the inputs into the derived properties).

Our next steps are to begin more detailed discussion and design. One thread of that work is to collect a database of interesting cases, both those handled well and those handled poorly by existing software, to act as a test dataset for spec development and as a resource for implementors. Contributions to this are very welcome.