
Subject: Re: What is the U++ way to replace a missing font glyph?

Posted by [mirek](#) on Mon, 22 Jun 2020 18:42:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

Oblivion wrote on Mon, 22 June 2020 17:20Hello Mirek,

Quote:

Generally, it would probably be more simple to just keep the list updated...

What I have in my mind can be summarized as follows:

- 1) Make the same list dynamic (using a vector maybe?)
- 2) Accordingly, add global functions to manipulate or query the list on demand.
- 3) Keep the existing code as intact as possible.

Basically, it is a global list management.

For example (just to give an idea)

```
void AddFallbackFont()
void RemoveFallbackFont()
void InsertFallbackFont()
void ResetFallbackFonts() // Always resets the list to U++ defaults.
const Vector<>& GetFallbackFonts() // Not sure about this one, but may be used to fill font lists in
UI, if needed;
```

Frankly I don't really care about emojis or icons. The "modern terminal emulation scene" is a very interesting and weird habitat, but what I really need is some special fallback fonts when a terminal connection requires font adjusting depending on the environment over a remote (SSH) connection, for example.

Besides U++ apps in general can benefit from this.

My main concern would be performance (though the impact of using a dynamic list should be negligible) and the potential concurrency problems these functions may bring in.

What do you think?

Best regards,
Oblivion

I think that whatever fonts you might add by AddFallbackFont, you can have in the list already... :)

I mean, how are you going to use this?

Mirek
