Subject: What is the U++ way to replace a missing font glyph?
Posted by Oblivion on Fri, 19 Jun 2020 11:33:44 GMT
View Forum Message <> Reply to Message

Hello,

I am working on a range-based and fallback font substitution mechanism for Terminal ctrl and I don't want to reinvent the wheel.

Is there already a way to replace missing glphs in U++ API?

For example, most fonts do not have the glyphs for braille alhabet or unicode emojis/icons. I want to use alternative fonts for missing glpyhs.

I see some methods for Font, suggesting that it is possible but I couldn't find a simple example.

I'd be grateful if anyone could clarify this point and/or post an example.

Edit: I see a static font replacement list in Draw/FontCR.cpp. Is it how U++ does this? If so, can we have a way to modify and let apps manage the list?

Best regards,
Oblivion

Subject: Re: What is the U++ way to replace a missing font glyph?
Posted by mirek on Mon, 22 Jun 2020 12:17:20 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Fri, 19 June 2020 13:33
Edit: I see a static font replacement list in Draw/FontCR.cpp. Is it how U++ does this?

Indeed.

Quote:
If so, can we have a way to modify and let apps manage the list?

We can talk about that...Generally, it would probably be more simple to just keep the list updated...

Mirek

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

Posted by Oblivion on Mon, 22 Jun 2020 15:20:09 GMT
View Forum Message <> Reply to Message

Hello 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 negligable) and the potential concurrency problems these functions may bring in.

What do you think?

Best regards,
Oblivion

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 negligable) 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

Subject: Re: What is the U++ way to replace a missing font glyph?
Posted by Oblivion on Mon, 22 Jun 2020 21:11:51 GMT
View Forum Message <> Reply to Message

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

I mean, how are you going to use this?

OK, if we can add some new faces to fallback fonts list, it would be sufficient and great. :)

For one, an important monospace font with great glpyh coverage is missing: FreeMono. (FreeSans and FreeSerif is already in the list but their monospaced counterpart is missing).

Also, FontAwesome should be in that list too. (It has extensive icons coverage, used by many apps.)

These two would improve the result of font glyph substitution mechanism of U++.

There are other font families such as Noto for extensive CJK, Devanagari, etc. coverage but they can be picked and added later, if needed.

As a side note, I think it would be better to have a Font constructor that takes a facename. (Can be used to construct font arrays from facenames, using the C++11 initializer lists.)

Best regards,
Oblivion

Subject: Re: What is the U++ way to replace a missing font glyph?
Posted by mirek on Fri, 03 Jul 2020 11:56:50 GMT
View Forum Message <> Reply to Message

I have added FreeMono. I do not have FontAwesome here to create its coverage (uppbox/FontCover).

As for facename as constructor parameter, I would rather not. I do not like to encourage the idea that facenames are shared across computers...

Mirek

Subject: Re: What is the U++ way to replace a missing font glyph?
Posted by Oblivion on Fri, 03 Jul 2020 12:18:57 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Coverage of FontAwesome;

"FontAwesome", 0xc0000000, 0x88000002,

As for facename as constructor parameter, I would rather not. I do not like to encourage the idea that facenames are shared across computers...

No worries, I can live without it.

Thank your very much!

Best regards,
Oblivion

Subject: Re: What is the U++ way to replace a missing font glyph?
Posted by mirek on Fri, 03 Jul 2020 14:21:35 GMT
View Forum Message <> Reply to Message

Hm, looks like cover is actually not used anymore....

In fact, looking at it, there should be some work done. Right now, replacements do not care about font style much...

Mirek
