

[Home](#) » [Uncategorized](#) » Fontlink for CJK on English Windows 10

UNCATEGORIZED

Fontlink for CJK on English Windows 10

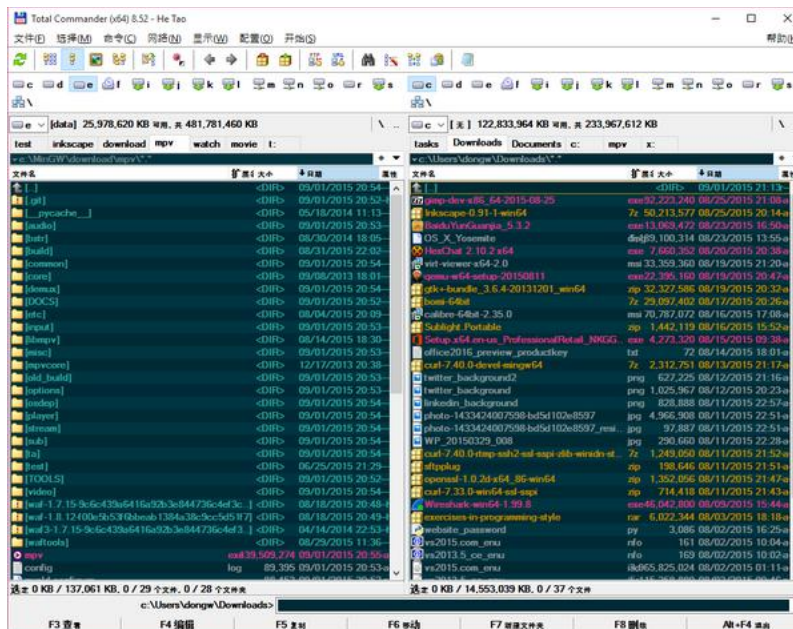
by [taohe](#) | Published [September 6, 2015](#) | [5 comments](#)

The issue

As in previous Windows 7/8/8.1, if the system locale is set to English, Chinese text would appear ugly in some applications.

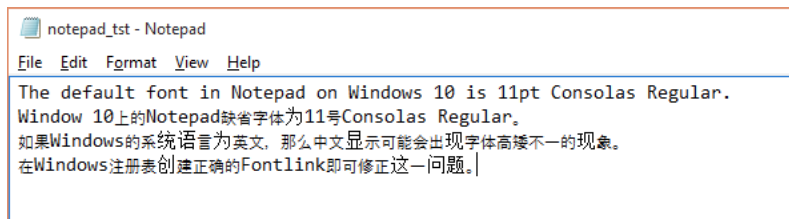
Recently, I upgraded to Windows 10 from Window 7. After the Windows 10 was activated, I did a clean installation. Then I had the same issue again. I thought it could be interesting to write a blog (my very first blog at [Shajisoft.com](#)) to record how I got rid of the issues.

A picture is worth one thousand words. The following is a screenshot of the main window of TotalCommander with the language switched to Chinese. Those Chinese text appears really ugly. Words are displayed in different heights/sizes respectively as if they were rendered by using different fonts.

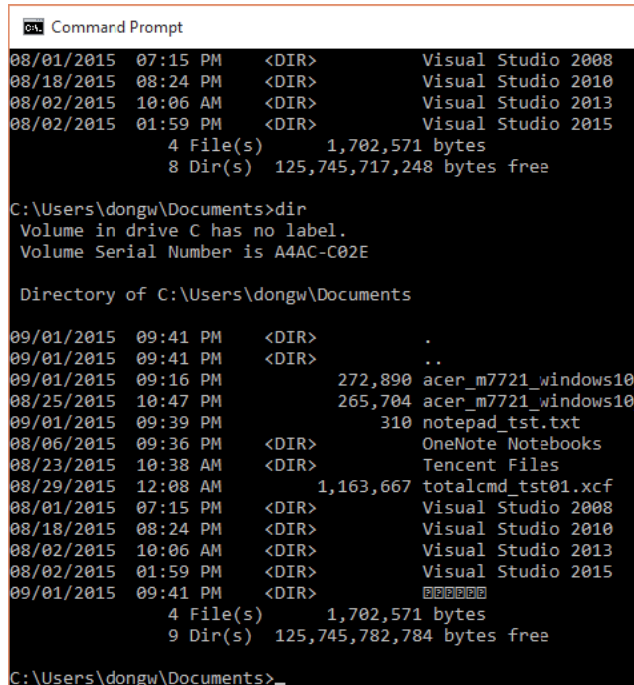


The Chinese interface of TotalCommander is displayed ugly on Windows 10 English by default.

The same issue happens to other applications including Windows system tools like Notepad and CMD, as shown below.

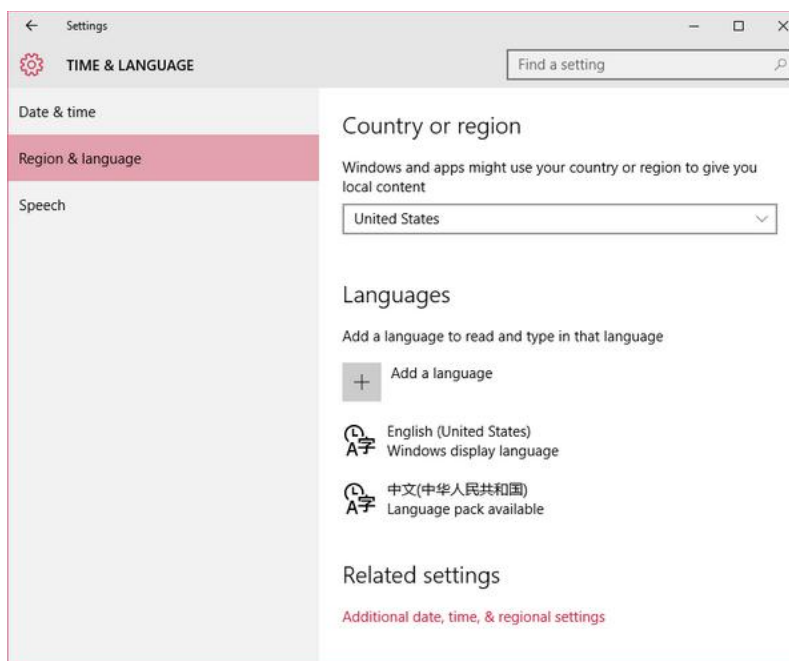


The default Notepad displays mixed Chinese and English on Windows 10.



The default CMD.exe fails to display Chinese directory name on Windows 10 English.

Just in case someone is curious, the default language of my Windows 10 box is English, as shown below



The regional and language configuration of the Windows used to create the above screenshots.

As shown above, Chinese text appears normal on the dialog of settings dialog. This means that the ugly display of Chinese text happens to some applications only. In addition, although the issue is with Chinese text, I believe the same issue would happen to Japanese and Korean text as well if the system default language is English.

Testing Environment

All the tests shown in this post were done on Windows 10 English x64 using the following applications:

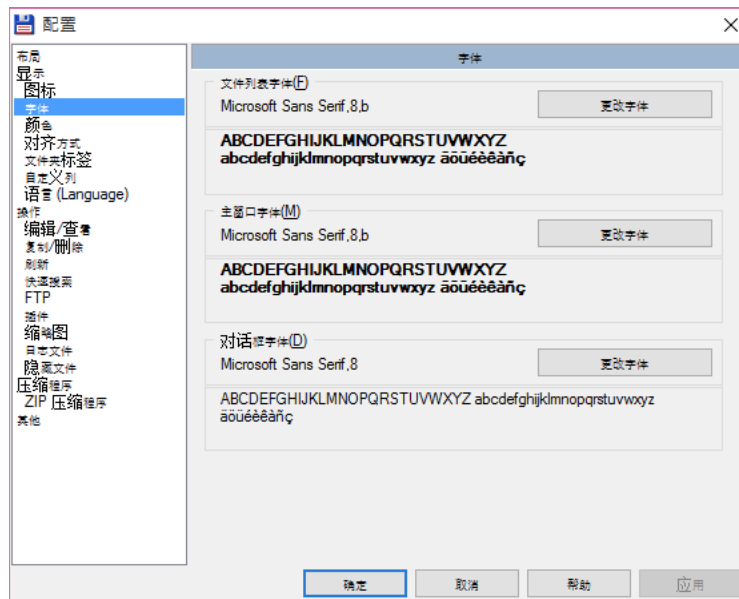
- TotalCommander 8.52 x64
- CMD (system default)
- Notepad (system default)

Analysis

Basically, the above screenshots show two issues, namely incorrect font or not font found to display Chinese text respectively. Windows makes use of several techniques to find font glyphs to display text. If an incorrect font is mapped, the display of the script would be abnormal. In cmd.exe, text rendering may be different, that is, if no proper font glyphs are found, the text would appear to be boxes. This usually happens to multi-byte text scripts in Chinese, Japanese, and Korean. So, to get around the issue, we need to map the fonts properly when needed.

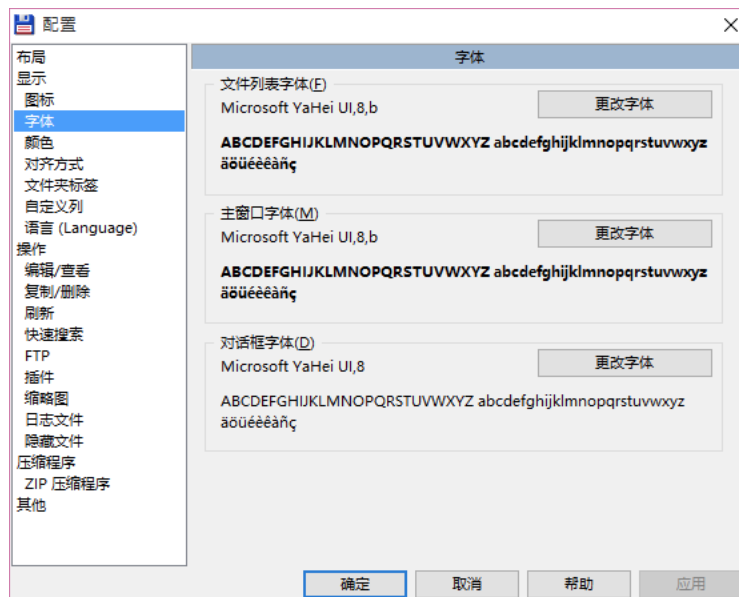
Solution 1 of 2: Select correct font

If applications like TotalCommander allow users to select fonts, we can select correct font which include the font glyphs needed to display the text, for instance Microsoft Yahei. In TotalCommander, the configuration of font can be modified on the configuration dialog. To open the configuration dialog, select menu Configuration->Options. The configuration of font is available on "Font" tab page. The default settings of the fonts after the language is switched to Chinese are as shown below:



The default font dialog in TotalCommander in Chinese language on Windows 10 English.

If we select a proper font, ie, Yahei UI, the Chinese text would be rendered normally, as shown below:



The font dialog in TotalCommander after setting font to Microsoft Yahei UI on Windows 10 English.

As shown above, once a correct font is select, the problem would be gone. Whereas, this solution is not optimal. In some cases, we still want to have English text rendered using a pretty mono font, while Chinese (or Japanese, Korean) text is still rendered properly, especially in applications like Notepad and CMD. In that case, we may need to set up proper font mapping which will be explained in the following section.

Solution 2 of 2: Set up correct fontlink in the registry

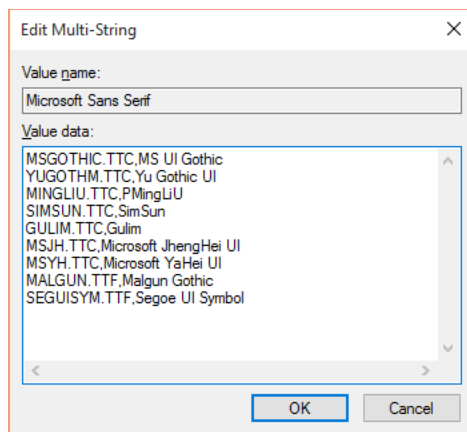
Fontlink is a virtual font technique used in Windows to link many fonts. A font can be linked to many other fonts. In this case, the font is called base font, the other fonts are linked fonts. We can set the application to use a proper base font mainly for English text, and set up the font link to allow proper rendering for texts in other language.

Many pages or posts can be found on Google if we search “Chinese English Windows text”, “fontlink fontlink meiryo ui”, or “fontlink yahei” etc. One can always refer to any of those on how to make changes in the registry on Windows for Fontlink. In this post, I am not going to show the steps in detail, but rather I’ll focus on the discussion on one of the important parameter called “scaling factor”, which is either not covered or explained unclearly in web pages.

Take TotalCommander as the example, the default font is Microsoft Sans Serif, as shown in the above screenshot. To allow proper rendering for Chinese text, all we need to do is to set up the proper fontlink. Fontlink is defined in Windows registry. To view or edit the registry key, run command “regedit”. Then browse to HKLM→Software→Microsoft→Windows NT→CurrentVersion→Fontlink→SystemLink.

NB: It's highly recommended to make a backup of the original key value by exporting the original registry key to a file by using the menu File→Export... in RegEdit.

The above registry key contains all fontlinks for some fonts which includes Microsoft Sans Serif. To view or modify fontlink for Microsoft Sans Serif, right-click on the entry, and select “Modify...”. The default configuration of the fontlink for Microsoft Sans Serif on Windows 10 English is as shown below:



The default fontlink values for Microsoft Sans Serif on Windows 10 English.

When looking for font glyphs to render text, Windows checks the list of linked fonts from top to bottom. To allow Chinese text properly rendered, we can add an appropriate font at the top of the list. Before making actual changes, let’s review the syntax of each line in the list defined in the fontlink.

Taking Microsoft Yahei UI as the example, the syntax of an item in the fontlink is as follows:

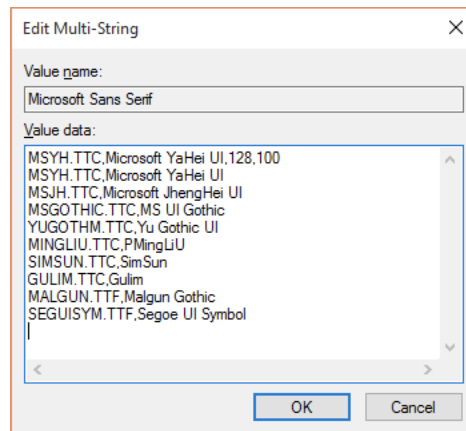
```
1 MSYH.TTC,Microsoft YaHei UI,128,96
```

The whole line is used by Windows when mapping font. Therefore, the following two lines are two different entries in terms of fontlink:

```
1 MSYH.TTC,Microsoft YaHei UI,128,96
2 MSYH.TTC,Microsoft YaHei UI
```

where “128,96” are known as scaling factors. According to [this link](#), the above two lines are for GDI and GDI+ respectively. That is, the first line will be picked up by GDI, while the second one is for GDI+. In order to add a new linked font which can be used by both GDI and GDI+, we ought to add two lines in the fontlink registry, one with scaling factors and one without respectively. TotalCommander uses GDI to render text, so we need to set up at least the fontlink with the two integers.

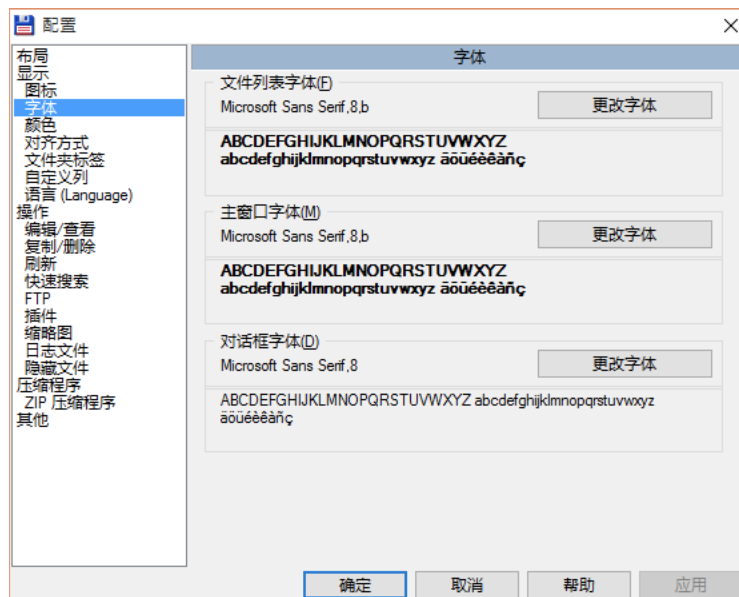
I can't find any useful information about how the scaling factors are used. But it seems to be true that if the two scaling factors are 128 and 96 respectively, the text won't be scaled. But in my case, the non-scaled Yahei UI looks a bit taller than Microsoft Sans Serif. I have to make the Chinese words appear shorter. Eventually, I set the fontlink as follows for TotalCommander:



The new settings of fontlink for Microsoft Sans Serif on Windows 10 English.

NB: To take effect of the new fontlink, sign out and in Windows again. There is no need to reboot the computer!

With the new fontlink configuration, as shown in the following screenshot, Chinese text rendered properly even though the select font is still Microsoft Sans Serif.



The font dialog of TotalCommander after setting fontlink for Microsoft

Sans Serif on Windows 10 English.

Now, I would like to elaborate a bit more on the scaling factor numbers. It's mysterious how the numbers are used. It's quite true that the default values of the two factors are 128 and 96 respectively. In other words, if the two default values, 128,96, are set in fontlink, there won't be scaling effect. But many web pages would suggest other values, which I've found is very confusing, especially some suggestions to set the factors to values by multiplying the default ones with the same number (ie, 1.6). Please read on for the reasons. I can't find any useful information about the details of the two numbers. The only one relevant reference is [a source file](#) of gdipp project where the two factors are used to compute a new scaling factor as follows

```
1 new_info.scaling = (factor1 / 128.0) * (96.0 / factor2);
```

This means that setting factors by multiplying the default values with the same value doesn't make any difference! With the above formula and some experiments, I think that we can draw the following conclusions regarding the two scaling factors:

1. To allow programing using GDI to take advantage of Fontlink, we need to add items with scaling factors in the registry.
2. The scaling factors may depend on the actual DPI. In other words, different display device may require different scaling factors in the fontlink registry.
3. The linked fonts are not necessarily in the same size of the base fonts. To increase or reduce the size of the linked font, it's possible to keep one of the scaling factor to the default value, and just change the other one. It doesn't matter which one has the default value.
4. Setting scaling factors by multiplying the default values (128 and 96) with the same value doesn't make any difference.

Still taking TotalCommander as the example, let's compare the results by setting the scaling factors to the following values respectively:

1. 128,100, and 122, 96 respectively. They should give the same results.
2. 128,128, and 64,64 respectively. They should give the same results.
3. 128,72, and 171,92 respectively. They should give the same results.

The following figure shows the screenshots of TotalCommander with different scaling factors used in fontlink.



Comparison of effects of various scaling factors in the configuration of fontlink on Windows 10 English.

On my 20 inch LCD monitor set to 1920×1080, using the default scaling factor, 128 and 96, Chinese text appears a little taller than English text in TotalCommander. I have to set the scaling factor either to "128,100" or "122,96". The default scaling factors work fine in Notepad and CMD.exe.

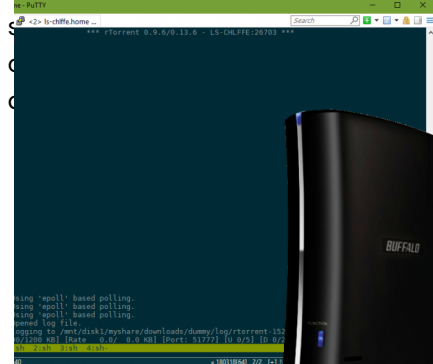
Summary

On Windows 10 English, Chinese, Japanese and Korean (CJK) text may be rendered by using a wrong font therefore appear ugly. There are basically two solutions to get rid of the issue, namely to set a correct font in the application or to set correct fontlink in the registry so that a correct font can be used for CJK text.

The second solution is usually preferred mainly for two reasons:

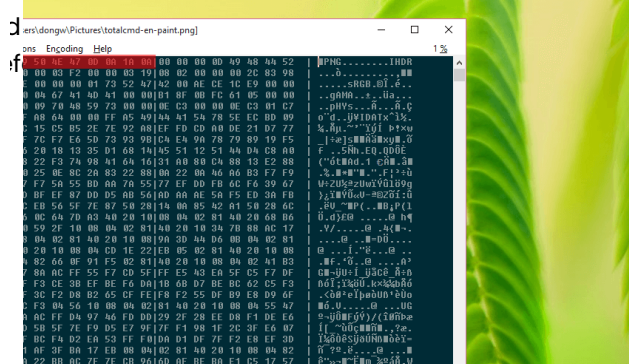
1. We still want English text to be rendered in the preferred font (ie, Consolas).
2. The application may not allow us to change the font.

Using fontlink technique, a base font is linked to many fonts. The linked fonts may not necessarily in the same size as the base font. It's necessary to scale the linked font in this case. It's mysterious how the scaling factors are used in Windows. But it has been shown that to scale up or down the linked font, we can



Published April 8, 2018

[Build rtorrent v0.9.6 for Linux on ARM \(armv5tel\) with SSL and TLSv1.2](#)



Published October 11, 2015

[Why the Same Screenshot is Saved as PNG of Different Sizes by Different Tools](#)

Your email address will not be published. Required fields are marked *

***COMMENT**

***NAME**

***EMAIL**

WEBSITE

POST COMMENT

5 thoughts on “Fontlink for CJK on English Windows 10”

[5 comments](#)



Frank

December 28, 2015, [12:38 am](#)

Thank you so much for this blog. I was having the same problems after upgrading to Windows 10 and almost gave up on it until I found your blog. My problems were fixed by following option 2 that you gave. Thanks again.

[REPLY](#)



Erika

January 7, 2016, [4:03 pm](#)

Thank you, your tips helped me solve the problem with Windows 10 on my pc, it was driving me crazy! Thanks

[REPLY](#)



Erich

June 16, 2016, [4:59 pm](#)

I created a multi string value “Consolas” with same value data set up as your “Sans Serif”. It works for me on both cmd prompt and notepad. thank you for so good idea!

[REPLY](#)



Erich

October 31, 2016, [9:49 am](#)

I have worked a batch to add fontlink for Consolas.
For your reference:

```
REG ADD "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Fontlink\SystemLink" /v  
Consolas /t REG_MULTI_SZ /d "MSYH.TTC,Microsoft YaHei UI,128,96" \0"MSYH.TTC,Microsoft YaHei  
UI" \0"Arial.TTF,Arial"
```

REPLY



taohe Post author

November 1, 2016, 8:40 am

Thanks for sharing the command line example.

REPLY



[WHY THE SAME SCREENSHOT IS SAVED AS PNG OF DIFFERENT SIZES BY DIFFERENT TOOLS >](#)

© 2022 [Shajisoft](#) – All rights reserved

Powered by [WP](#) – Designed with the [Customizr theme](#)

