Insights and Rants

News and general opinion, often privacy, security or computer related, but could be about anything really, including religion, politics, the environment, business or audio books. "Never ascribe to malice, that which can be explained by incompetence/stupidity/neglect." ~ Hanlon's Razor

Saturday, October 14, 2023

Clarion Fixer (Part 1): Using the App

One of my pet peeves with Clarion is that the applications it generates by default look like they were written for Windows 95, and not for current versions of Windows. In modern versions of Windows, Microsoft has "fixed" the old look by making all the buttons and controls fat, but Clarion insists on using the "Microsoft Sans Serif" font that was introduced in 1997.

It looks old, especially since I have already used and discarded Windows and Subsonic, for much the same reason. I never liked Acrid (1982). it shipped with Windows 3.2 and has been ugly ever since. I could not stand "MS Sans Serif" in a lineup from a DTG version of Windows. I want my programs to look like they were written recently, not 30 years ago.

Yes, I know, I know, I can "fix" the Clarion generated applications by editing the .ico files or using ini-file tweaks, but I have discovered that I could use a tiny utility called Search and Replace to fix the Clarion styles and templates. So when I create the app, I don't have to waste time fiving up them all over the place. This got me thinking: can I automate this? Whenever I install new or updated accessories for Clarion, I don't want to have to fix up the fonts yet again.

Introducing **clFixer**, which is freeware. It's not signed, and some anti-virus engines on VirusTotal think it's a trojan, but I think they are getting false positives. Naturally, it is written with Clarion 11.1, and uses the CapeSoft StringTheory (CBD) and WinEvent 3 (BSD) accessories. If you have them, you can download and modify the code here. The .exe file doesn't contain any exe or dll files. The zip file contains those. If not, you can read the .txt files and download the files. Extract the zip file into a suitable folder to try it out. The code is also available on GitHub.

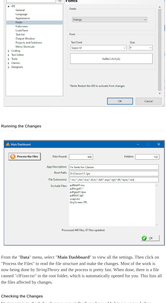
The first thing you need to do is make a copy of your Clarion folder, so you can see what changes have been made, and how they work. I have copied mine to "d:\Clarion11.1ga".

Exploring clFixer

Run the **clFixer** app, and from the main menu choose "**Data**" -> "Application Settings".

Click on the ellipse to the right of the "Root Folder" field:

Select the backup folder and click "OK". Note there is a list of "Exclude Files". Some changes, such as the "Concise New" change, will cause trouble if you don't exclude files, so you can decide whether a particular action must avoid modifying these files. Click "OK".



From the "**Data**" menu, select "Replacement Actions". This is a list of actions to be performed on files. All of these replacement actions are performed on files in the "Root Folder" and subfolders, matching the file name extensions

```
*.inc;*.clw;*.txa;*.dctx;*.def;*.equ;*.tpl;*.ff;*.tpw;*.red;
```

If you add an action to modify "readme.txt" it will **not** be modified because "txt" is not one of the file name extensions listed.

General Actions

Here is the first and one of the most generic actions. It applies to all files, except a small number on the "Exclude Files" list.

This is even more general because it applies to the excluded files as well as any other file that still mentions "MS Sans Serif".

Specific Actions

This action is specific to one file, and fixes up a form where the font names wrapped around over 2 lines.

This action is specific to the TTT extension only. It fixes a general replacement from "Grid" to "Input Ctl" intended for forms, and changes the font to "Calibri" for a setting to do with reports. If you use a wild card like this in the file name, be sure to omit the "|" after the wild card, and use only one file name. The software isn't expecting multiple file names in the field.

Line-Based actions

Sometimes a search and replace across an entire file is not what we had in mind, and can cause havoc. Here is an example. The term "#SHEET,DO,end" appears in multiple places, so I only want to change this specific one. Note that I have included the first part of the next line as well, to prevent the same replacement happening again if the file is scanned more than once.

Here is another example of a line-based replacement. You may wish to disable it altogether because your IDE font settings will specify the font anyway if you force them set as follows:

Running the Changes

From the "**Data**" menu, select "**Main Dashboard**" to view all the settings. Then click on "Process the Files" to read the file structure and make the changes. Note that the work is now being done by StringTheory and the process is pretty fast. When done, there is a file named "clFixer.log". In the root folder, which is automatically opened for you. This lists all the files affected by changes.

Checking the Changes

It's important to check the changes, especially if you have added in new ones. before clFixer saves a changed file it checks to see if a "**before**" file exists. This is a file with the same name as the file, but with the original file then was renamed, and will still have the original date/ time.

Here is a file comparison using the ExamDiff Pro utility, but there are others you may prefer. The original file is at the bottom, and the new file is at the top. It's important to do this because you don't want to screw up your actual working environment and have to reinstall things.

Once you apply the changes to your working environment, remember that source of your existing apply will have changed. Only **new** forms, reports and apps, and template-generated files will be affected by the new Clarion build.

Why pay use for a useful IDE for then free/Free product? My main objection is that this product allows the user to make bad font choices, and select something horrible like "Comic Sans". Shameful. Also, fonts have different widths and spacing, and I don't want text in Windows to be truncated because someone is trying to fit it into the space intended for Arial Narrow. Legibility is more important than style.

In the next part, we will take a look at some of the programming techniques (good and bad) used in this application.

Update 20 October 2023: Following feedback and further insights from Geoff Robinson and Christofolli, the download is updated to version 0.2

[Clarion Fixer (Part 1): Using the App]
[Clarion Fixer (Part 2): The Code]
[Change Mode vs View Mode]
[Clarion Build Events]
[Introducing Clarion FileManager (Part 1)]
[Introducing Clarion FileManager (Part 2)]
[**Clarion Resources** and Products]

Logo credits: Toolbox icon created by Rosenfeld.com Premium - Flaticon

Posted by Dorin Edwards at 1:26 PM Share

Labels: CapeSoft, Clarion, clFixer, Fonts, Productivity, StringTheory, WinEvents

No comments:

Post a Comment

Home Older Post

Subscribe to: Post Comments (Atom)

Copyright © 2006-2022 by Dorin Edwards. Off topic or spam comments will be deleted. Powered by Blogger.

Useful Links

Clarion Resources and Products
Installing Clarion 11 on Windows 10
Introduction to Clarion 11
Getting Started with Clarion (Part 1)
Getting Started with Clarion (Part 2)
Backing up your Clarion Projects
Learning Clarion (Part 1)
Learning Clarion (Part 2)
Learning Clarion (Part 3)
Learning Clarion (Part 4)
Learning Clarion (Part 5)
Learning Clarion (Part 6)
Learning Clarion (Part 7)
Adding Extensions to Clarion
Clarion FileManager (Part 1)
Clarion FileManager (Part 2)
Change Mode vs View Mode
Fact/Reviews - dirtcap web site still under construction
Jubilee Evans Art Studio
Jargon Free Guide to Computer Security
Lookup SA SMS numbers
Black and White Inc
Lifetime Movie Radio, also known as LM Radio

Contact Details

Dorin Edwards
email@dorin.ca
+1 (613) 831-0241
Skype: dorin_edwards

Contact information is private and confidential. Do NOT use my email address, phone number or Skype handle to add me to your mailing list, or for non-paid and unsolicited marketing messages. You can follow me on Twitter @dorinedwards or on Instagram as dorinedwards.