VCL

🏠 📄 🔍 ☰

Suchst Du die deutschsprachige DP?

Delphi-PRAXiS
The International Developer Community

Existing user? Sign In ▼   **Sign Up**

☰ **BROWSE** ▼   📢 **ACTIVITY** ▼   Search...🔍

Home ❯ Delphi Questions and Answers ❯ VCL   📁 **All Activity**

Looking for Icon Fonts support in Delphi for High-DPI and Themed app?

◀ imagelist   # Looking for Icon Fonts support in Delphi for High-DPI and Themed app?   ❶ Sign in to follow this   Followers   3

By Carlo Barazzetta, August 28, 2020 in VCL

◀ icons   ◀ vcl   ◀ fmx   ◀ high-dpi   ◀ vcl-styles

Ignore this topic

Carlo Barazzetta

Posted August 28, 2020   ⤴

IconFontsImageList components by @Carlo Barazzetta could be the answer: you can explorer the complete wiki to see how it works.

Members
➕ **85**
39 posts
Delphi 10.4 Sydney

Summary of library:

- An IconFontsImageCollection component that inherits from Delphi's CustomImageCollection and is compatible with VirtualImageList
- A IconFontsVirtualImageList, to use with Delphi version older than 10.3
- A rendering engine of Icon-fonts using GDI+ (from Delphi XE4)
- A complete backward compatibility with older Delphi versions (from Delphi 7)
- A useful Collection and Component editor, with support for Category of Icons
- A custom CharMap viewer, to easily select icons contained in any Font

- Support for changing the Color based on the active VCL Style.
- High performance of drawing engine
- Support for FMX (also for mobile platforms)
- It's free and open-source

Icons based on Fonts are a good alternative to bitmaps because they need only the Font installed in the system to obtain thousands of images (like the "Material Design Font Desktop.ttf" font: https://github.com/Templarian/MaterialDesign-Font). The icons scales perfectly, so, you don't need to multiple resolutions of your images to match the DPI of the monitors and multiple colors for Theme used.
The Collection of Icons can be rendered by a single Font/Color defined at collection level, or by different Fonts/Color defined at Icon level, so you can mix different icons from different Fonts in a single collection.

The library is quite stable, but any contribution is welcome!

❤ 5   🏆 1
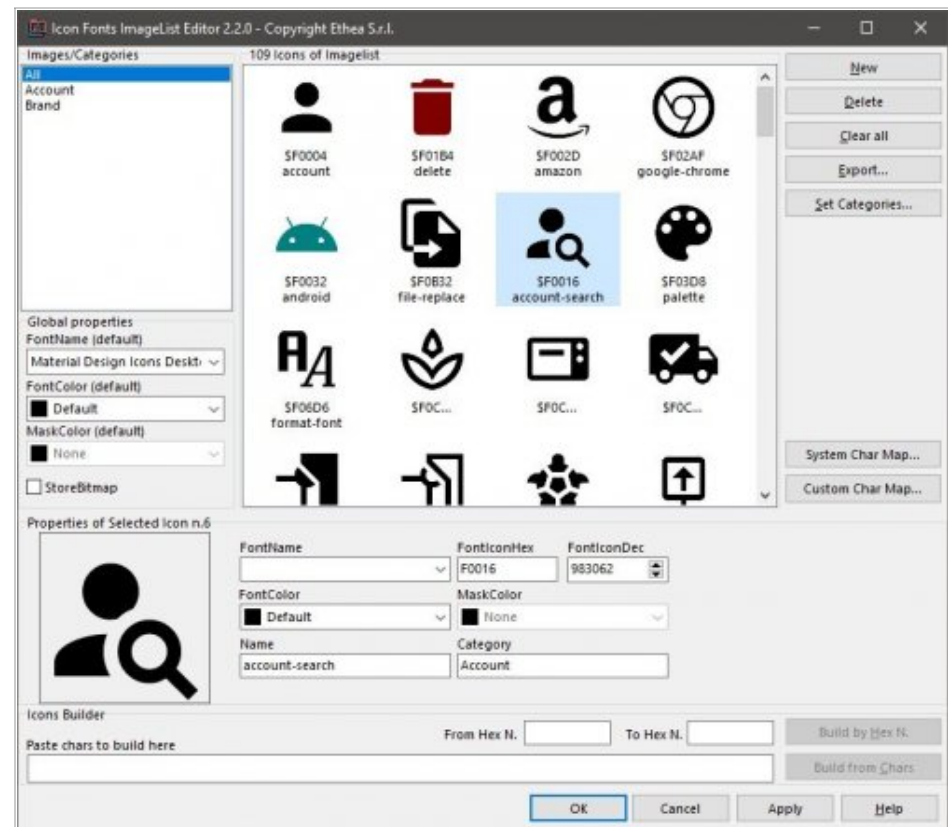
Attila Kovacs

Posted August 28, 2020 (edited)

Members

➕ 471

1477 posts

Are you sure that the font has to be installed?

I'm loading a custom font from the resources and it's working fine:

```
// Load custom Fonts before creating any forms
ResStream := TResourceStream.Create(hInstance, 'MONTSEF
try
  AddFontMemResourceEx(ResStream.Memory, ResStream.Size
finally
  ResStream.Free;
end;
```

Edited August 28, 2020 by Attila Kovacs

Carlo Barazzetta

Members

➕ 85

39 posts

Delphi 10.4 Sydney

Posted August 29, 2020

> **⌄ On 8/28/2020 at 9:32 PM, Attila Kovacs said:**
>
> Are you sure that the font has to be installed?
>
> I'm loading a custom font from the resources and it's working fine:
>
> ```
> // Load custom Fonts before creating any forms
> ResStream := TResourceStream.Create(hInstance, 'MON
> try
>   AddFontMemResourceEx(ResStream.Memory, ResStream.
> finally
>   ResStream.Free;
> end;
> ```

Yes, to develop the app the font must be installed on the system in order to use it in the IDE.

Deploying the app you can do as you explain: I have similar code in the demo, which responds to the OnFontMissing event raised by the component if Font is not installed:

```
//The "material desktop font is not installed into system
LFontFileName := ExtractFilePath(Application.ExeName)+'..
if FileExists(LFontFileName) then
begin
  {$IFNDEF D2010+}
  AddFontResource(PChar(LFontFileName));
  {$ELSE}
  AddFontResource(PWideChar(LFontFileName));
  {$ENDIF}
  SendMessage(HWND_BROADCAST, WM_FONTCHANGE, 0, 0);
end
```

❤ 1

Ondrej Kelle

Members
⊕ **56**
73 posts

Posted August 29, 2020

> ### ⊘ On 8/29/2020 at 10:54 AM, Carlo Barazzetta said:
>
> Yes, to develop the app the font must be installed on the system in
> order to use it in the IDE.
> Deploying the app you can do as you explain:

If you wanted to avoid the requirement of having the fonts installed
globally in Windows you could write a design package with the fonts linked
in as resources. The package, when loaded by the IDE, could load the fonts
from the resources (using the code shown by @Attila Kovacs) and call
Screen.ResetFonts to signal the IDE to reinitialize its font list . After that the
fonts would be available to the IDE.

pyscripter

Members
⊕ **479**

Posted August 29, 2020 (edited)

> ### ⊘ On 8/29/2020 at 2:31 PM, Ondrej Kelle said:
>
> If you wanted to avoid the requirement of having the fonts installed
> globally in Windows you could write a design package with the fonts
> linked in as resources. The package, when loaded by the IDE, could

592 posts
Delphi 10.3 Rio

load the fonts from the resources (using the code shown by
@Attila Kovacs) and call Screen.ResetFonts to signal the IDE to
reinitialize its font list . After that the fonts would be available to the
IDE.

Indeed:
From https://docs.microsoft.com/en-us/windows/win32/api/wingdi/nf-wingdi-addfontmemresourceex

> ### ⊘ Quote
>
> This function allows an application to get a font that is embedded in a
> document or a webpage. A font that is added
> by AddFontMemResourceEx **is always private to the process that
> made the call and is not enumerable**.
> A memory image can contain more than one font. When this function
> succeeds, pcFonts is a pointer to a DWORD whose value is the number
> of fonts added to the system as a result of this call. For example, this
> number could be 2 for the vertical and horizontal faces of an Asian
> font.
> When the function succeeds, the caller of this function can free the
> memory pointed to by pbFont because the system has made its own
> copy of the memory. To remove the fonts that were installed,
> call RemoveFontMemResourceEx. However, when the process goes
> away, the system will unload the fonts even if the process did not
> call RemoveFontMemResource.

Edited August 29, 2020 by pyscripter

Carlo Barazzetta

Posted August 30, 2020   ⤳

Members
➊ 85
39 posts
Delphi 10.4 Sydney

Actually the components do not install any fonts on purpose, to avoid
thinking they only works with a specific font ... The demo uses a sample
font, and also shows how to load it on-the-fly if it is not installed on the
system.
The downside is that when you open the demo you don't see the icons
right away in the IDE and this could make you think that the component is
not working properly...

The other aspect is that any font loaded as a resource in the IDE affects the possibility of installing a new version on the same font in the system...
I think that letting the developer choose to install the fonts he needs to use in his applications to use them with the components is the right and least invasive choice, don't you agree?

Carlo Barazzetta

Members
 85
39 posts
Delphi 10.4 Sydney

Posted September 6, 2020

The transition to "Virtual" is now complete (released ver.2.3).
Demos for 10.3 and 10.4 now uses native TVirtualImageList + TIconFontsImageCollection, and the TIconFontImage component can also be linked to VirtualImageList and IconFontsVirtualImageList (not only to IconFontsImageList).
I'm searching for users and contributors to the project, expecially for older Delphi versions (because the components are compatible from Delphi 7 to current): it's very hard to me to maintain backward compatibility without help.
From Delphi XE4 they uses GDI+ to render icons.

Carlo Barazzetta

Members
 85
39 posts
Delphi 10.4 Sydney

Posted September 16, 2020

I've published an article about SVGIconImageList and IconFontsImageList:
https://ethea.it/icons_in_delphi/

❤ 2   🏆 1

aehimself

Members
 273
719 posts
Delphi 10.4 Sydney

Posted August 23, 2021

Is it just me, or even the demo application does not import the font file properly when started for the first time?
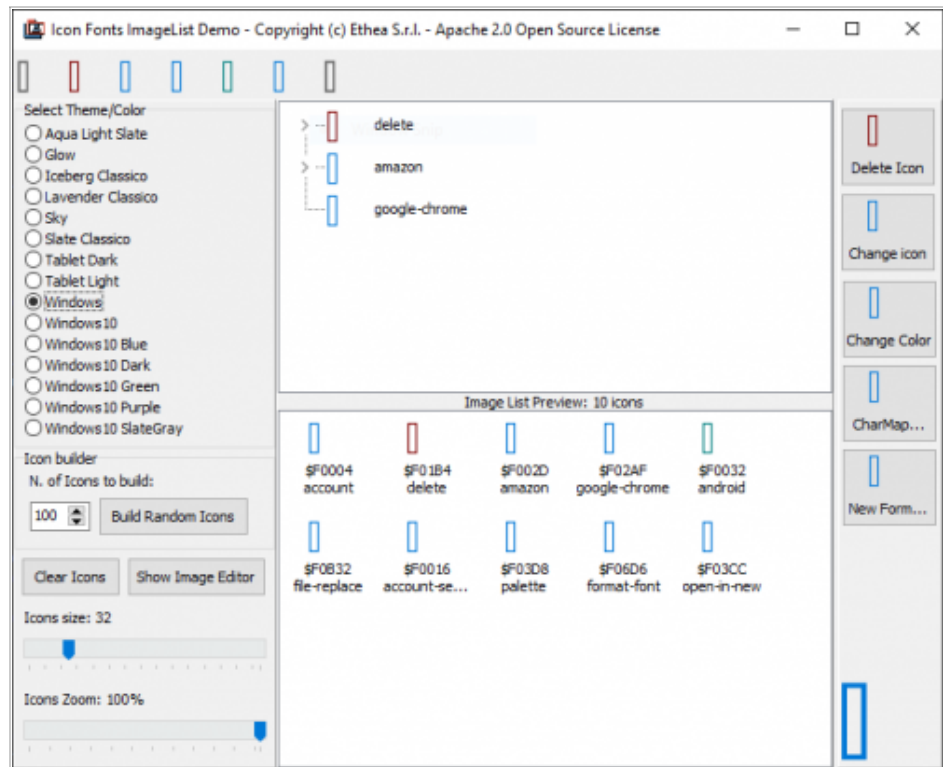
If the font is installed on the system (manually or by AddFontResource due to the first run) everything appears correctly.

I tried Screen.ResetFonts and ImageList.RecreateBitmaps after the SendMessage, without success. Dumping everything on the disk with .SaveToPngFiles also saves the rectangles only.

I'm using the latest snapshot from Git directly.

Carlo Barazzetta

Members
 85
39 posts
Delphi 10.4 Sydney

Posted August 23, 2021

When starts, the demo looks for the font installed into Windows: if not try loading from disk in this event of the datamodule:

```
procedure TdmImages.IconFontsImageCollectionFontMissing(
   const AFontName: TFontName);
var
   LFontFileName: string;
begin
   inherited;
   //The "material desktop font is not installed into system
   LFontFileName := ExtractFilePath(ParamStr(0))+'..\Fonts\M
   if FileExists(LFontFileName) then
```

```
      begin
        {$IFNDEF D2010+}
        AddFontResource(PChar(LFontFileName));
        {$ELSE}
        AddFontResource(PWideChar(LFontFileName));
        {$ENDIF}
        SendMessage(HWND_BROADCAST, WM_FONTCHANGE, 0, 0);
      end
      else
      begin
        //If the font file is not available
        raise Exception.CreateFmt('Warning: "%s" font is not pr
          'Please download at https://materialdesignicons.com a
            [AFontName]);
      end;
  end;
```

If the call to AddFontResource fails on your machine I don't known why...
please use the issue section of the project to continue discussion...
https://github.com/EtheaDev/IconFontsImageList/issues
Another reason could be that the font is installed only as resource of the
IDE: I reccommend to install it manually in the developer machine as system
font.
bye
Carlo

Carlo Barazzetta

Members
✪ 85
39 posts
Delphi 10.4 Sydney

Posted August 24, 2021 (edited)   ⤲

After several attempts we found these problems:
1) after calling AddFontResource and notified in windows that a font has
been added calling SendMessage(HWND_BROADCAST, WM_FONTCHANGE,
0, 0); the application hangs for several seconds waiting for all processes to
respond to the message... 😣
Using PostMessage(HWND_BROADCAST, WM_FONTCHANGE, 0, 0); instead
of SendMessage does not hang.
https://social.msdn.microsoft.com/Forums/en-US/6900f74f-6ece-47da-
88fc-f9c8bcd40206/sendmessage-api-slow?forum=wpf

2) In any case the imagelist does not draw icons correctly when it is using
GDI+: only by adding a sleep (500) you get the correct drawing of the icons.
It seems that calling AddFontResource does not immediately update the list

of fonts available to GDI+

For now the solution is to "wait" 500 millisecond at startup application after adding font loaded from disk:

```
AddFontResource(PWideChar(FFontFileName));
PostMessage(HWND_BROADCAST, WM_FONTCHANGE, 0, 0);
//Wait for Font available on GDI+ collection for drawir
Sleep(500);
```

But is not an elegant solution... Any idea?

Another article with similar problem:
https://www.vbforums.com/showthread.php?883637-GDI-problems-with-GdipCreateFontFamilyFromName

Edited August 24, 2021 by Carlo Barazzetta

Anders Melander

Members
 ⊕ **1151**
1655 posts
Delphi 11 Alexandria

Posted August 24, 2021

> **⊙ On 8/24/2021 at 10:35 AM, Carlo Barazzetta said:**
>
> After several attempts we found these problems

Maybe this is a clue that you're using fonts for something that they were not intended for...

aehimself

Members
 ⊕ **273**
719 posts
Delphi 10.4 Sydney

Posted August 24, 2021

> **⊙ On 8/24/2021 at 1:03 PM, Anders Melander said:**
>
> Maybe this is a clue that you're using fonts for something that they were not intended for...

You mean a font is not meant to be installed runtime or not to be drawn with GDI+?

Anders Melander

Posted August 24, 2021

Members
⊕ 1151
1655 posts
Delphi 11 Alexandria

> **⊘ On 8/24/2021 at 1:15 PM, aehimself said:**
>
> You mean a font is not meant to be installed runtime or not to be drawn with GDI+?

No. Here's some more clues:
https://en.wikipedia.org/wiki/Font
https://docs.microsoft.com/en-us/windows/win32/gdi/fonts-and-text

Carlo Barazzetta

Posted August 24, 2021

Members
⊕ 85
39 posts
Delphi 10.4 Sydney

> **⊘ On 8/24/2021 at 1:03 PM, Anders Melander said:**
>
> Maybe this is a clue that you're using fonts for something that they were not intended for...

I didn't understand what you mean...
I am using "fonts of icons" to draw bitmaps in an ImageCollection / ImageList, to get scaled images for any DPI... I am using GDI + for best performance and transparency support ...
If the font is already installed in the system it works fine, the problem is only when deploying an application without installing the font during setup: only in this case the application must load and add the font at startup, before use it to draw bitmaps, but I don't know why I need to add a sleep 😩

Lars Fosdal

Posted August 24, 2021

Have you checked if your app receives any notification after the installation of the font?
A sleep seems so random... what if the system is REALLY busy so that the sleep is too short?

Administrators
⊕ 1268
2430 posts
Delphi 11 Alexandria

aehimself

Posted August 24, 2021

Members

 273

719 posts

Delphi 10.4 Sydney

> ⊗ **On 8/24/2021 at 3:41 PM, Lars Fosdal said:**   ⤤
>
> Have you checked if your app receives any notification after the installation of the font?

There is no automatic notification as far as I'm aware, but the app receives the WM_FONTCHANGE message which was broadcasted. In the discussion on GitHub I mentioned that if I put a label on the form, set the font name to the same as in the ListView and paste some characters it shows up correctly without PostMessage / Sleep. This makes me believe that the installation is successful, only GDI+ needs some time before it can access a freshly installed font (or implementation is not correct). Unfortunately I have zero experience with any graphic-related stuff so can not confirm / debug :(

> ⊗ **On 8/24/2021 at 3:41 PM, Lars Fosdal said:**   ⤤
>
> A sleep seems so random... what if the system is REALLY busy so that the sleep is too short?

This is why it's not even viable as a workaround imo.

Create an account or sign in to comment

You need to be a member in order to leave a comment

Create an account

Sign up for a new account in our community. It's easy!

Register a new account

Sign in

Already have an account? Sign in here.

Sign In Now

Looking for Icon Fonts support in Delphi for High-DPI and Themed app?

Terms of Service   Privacy Policy   Code of Conduct   Imprint