

User Interface - Feature #3876

add support for "direct font control" at the widget level

01/11/2019 04:21 PM - Greg Shah

| | | | |
|--|--------------|------------------------|------------|
| Status: | Closed | Start date: | |
| Priority: | Normal | Due date: | |
| Assignee: | Hynek Cihlar | % Done: | 100% |
| Category: | | Estimated time: | 0.00 hour |
| Target version: | | vendor_id: | GCD |
| billable: | No | | |
| Description | | | |
| Related issues: | | | |
| Related to User Interface - Bug #4377: Custom widget font does not influence ... | | | New |
| Related to User Interface - Bug #4611: Implement a font enumeration Java API ... | | | New |

History

#1 - 01/11/2019 04:24 PM - Greg Shah

The idea is to implement FWD extension attributes which allow arbitrary fonts to be used based on common font names/font family names, styles, sizes and so forth. This would be completely independent of the FONT TABLE approach. Any widget would support these attributes similar to how we have BG/FGCOLOR-RGB support.

Dynamic layout ([#3875](#)) should take the font metrics into account so that widgets using larger fonts will "want" more space.

#2 - 01/13/2019 09:17 AM - Hynek Cihlar

Note to the implementor. There is already some support implemented in TREEVIEW widget, see its FONT-NAME and FONT-SIZE attributes and their implementations.

#3 - 01/14/2019 07:40 AM - Constantin Asofiei

GuiFontResolver should be changed to consider the FONT-NAME and FONT-SIZE (and any other font settings), to resolve a FontCache instance with the actual font details. Currently, only the font ID from the font-table is used - and this is translated to a FontCache which in turn is passed to the GUI drawing routines.

I don't think we need major widget-level drawing code changes (just checks that the explicit font is considered); but the metrics-related APIs like AbstractWidget.getTextWidth will need to be changed to consider the font attributes.

#4 - 02/19/2019 10:42 AM - Hynek Cihlar

- Status changed from New to WIP

#5 - 02/19/2019 10:43 AM - Hynek Cihlar

- Assignee set to Hynek Cihlar

#6 - 02/19/2019 10:46 AM - Hynek Cihlar

I took this issue to resolve custom font assignment to TREEVIEW widget needed in #3750. I did a preliminary work for custom font assignment in 3750b, but it only works for TREEVIEW, and it has issues. The proper custom font support will be relatively trivial.

#7 - 02/19/2019 10:46 AM - Hynek Cihlar

- % Done changed from 0 to 50

#8 - 04/23/2019 03:53 PM - Hynek Cihlar

This issue is resolved with revision 11306 in 3767a. Please review.

#9 - 06/11/2019 02:10 PM - Hynek Cihlar

The changes for this issue were delivered with 3767a in trunk as revision 11315.

#10 - 06/11/2019 02:11 PM - Hynek Cihlar

- % Done changed from 50 to 100

No work is remaining for this issue.

#11 - 06/11/2019 02:50 PM - Greg Shah

- Status changed from WIP to Closed

#12 - 06/14/2019 10:03 AM - Hynek Cihlar

Created task branch 3876a and checked in a fix for missing handle.unwrapCustomFont method in trunk revision 11315. This method is needed when converting an app that references any of the CustomFont attributes on a handle. Please review.

#13 - 06/14/2019 10:13 AM - Greg Shah

Code Review Task Branch 3876a Revision 11317

I'm fine with the change.

Don't we also need to add CustomFont to HandleCommon?

#14 - 06/14/2019 10:29 AM - Hynek Cihlar

Greg Shah wrote:

Code Review Task Branch 3876a Revision 11317

I'm fine with the change.

Don't we also need to add CustomFont to HandleCommon?

True. Added in revision 11318. Please review.

#15 - 06/14/2019 10:31 AM - Greg Shah

Code Review Task Branch 3876a Revision 11318

In HandleCommon, please move CustomFont into the FWD extension section at the bottom.

#16 - 06/14/2019 10:33 AM - Hynek Cihlar

Greg Shah wrote:

Code Review Task Branch 3876a Revision 11318

In HandleCommon, please move CustomFont into the FWD extension section at the bottom.

Done.

#17 - 06/14/2019 11:30 AM - Greg Shah

Code Review Task Branch 3876a Revision 11319

I'm good with the code. Please merge to trunk.

#18 - 06/14/2019 11:58 AM - Hynek Cihlar

3876a merged to trunk as revision 11317, and archived.

#19 - 11/06/2019 08:10 AM - Greg Shah

- Related to Bug #4377: Custom widget font does not influence the widget size and doesn't propagate from frame down to widgets added

#20 - 04/10/2020 08:16 AM - Greg Shah

I've just edited the [Font Control in 4GL Widgets](#). Please review and let me know if there is anything inaccurate.

Some changes to make:

- What is the format of the font name? Is it a font family name? Show some examples in the attributes table.
- Is there any support for using a "generic" serif or sans-serif font family?
- What happens if the point size being set is invalid or unknown?
- Please add a 4GL code example.

Questions:

- Why do we only support fonts listed in the directory? We would really want to be able to reference any vector font that is registered in the system and which is usable from Java. There should be no need for a static definition.
- A programmer would probably need the ability to list out the fonts and to get some details of the sizes/styles/attributes of each font. By attributes, I mean things like serif/sans-serif.

#21 - 04/10/2020 09:00 AM - Hynek Cihlar

Greg Shah wrote:

I've just edited the [Font Control in 4GL Widgets](#). Please review and let me know if there is anything inaccurate.

Some changes to make:

- What is the format of the font name? Is it a font family name? Show some examples in the attributes table.

In fact this is either the system font family name (available to Java runtime or the web browser in case of Web GUI) or the font name as defined in the server directory. I will provide some examples and more information in [Font Control in 4GL Widgets](#).

- Is there any support for using a "generic" serif or sans-serif font family?

These font faces are supported by both the Java runtime and the supported web browsers. I will also provide some examples for these.

- What happens if the point size being set is invalid or unknown?

If any of the attribute values are invalid or the font can't be materialized for any other reason, the logic will fall back to the standard 4GL font resolution. I will add this info to [Font Control in 4GL Widgets](#).

- Please add a 4GL code example.

Will do.

Questions:

- Why do we only support fonts listed in the directory? We would really want to be able to reference any vector font that is registered in the system and which is usable from Java. There should be no need for a static definition.

Actually we do. I didn't realize this when I wrote the initial version of [Font Control in 4GL Widgets](#).

- A programmer would probably need the ability to list out the fonts and to get some details of the sizes/styles/attributes of each font. By attributes, I mean things like serif/sans-serif.

He could do this using the standard system tools. Or do you want to add other 4GL extensions that would allow this?

#22 - 04/10/2020 09:07 AM - Greg Shah

He could do this using the standard system tools. Or do you want to add other 4GL extensions that would allow this?

4GL extensions are not needed, as long as one can call Java code for the inspection. In other words, if you can use our direct Java access from 4GL code to do the inspection, then nothing else is needed. If this is possible, please add a section to the wiki page to explain the idea and show an example.

#23 - 04/10/2020 09:12 AM - Hynek Cihlar

Greg Shah wrote:

He could do this using the standard system tools. Or do you want to add other 4GL extensions that would allow this?

4GL extensions are not needed, as long as one can call Java code for the inspection. In other words, if you can use our direct Java access from 4GL code to do the inspection, then nothing else is needed. If this is possible, please add a section to the wiki page to explain the idea and show an example.

For the Swing driver this will work. But for the Web GUI driver I would also have to add support to the driver to provide the available fonts and create a Java API available to the legacy code.

#24 - 04/10/2020 09:13 AM - Greg Shah

Please create a new task for that. We will defer work on it for now. In the wiki page, explain the idea and link to that task as a possible future enhancement. Also make the task Related to this one.

#25 - 04/10/2020 09:22 AM - Hynek Cihlar

- Related to Bug #4611: *Implement a font enumeration Java API directly callable from legacy code* added

#26 - 04/10/2020 09:23 AM - Hynek Cihlar

Greg Shah wrote:

Please create a new task for that. We will defer work on it for now.

Done.

#27 - 04/16/2020 06:19 PM - Hynek Cihlar

I added the required information in [Font Control in 4GL Widgets](#) together with an example. I found a couple of issues with the font control along the way, these are resolved in 4231b revision 11464.

Also in the example I tried to use direct Java:

```
using java.awt.* from java.  
using java.lang.* from java.def var gEnv as GraphicsEnvironment.  
def var availableFonts as String extent.  
gEnv = GraphicsEnvironment.getLocalGraphicsEnvironment().  
availableFonts = gEnv.getAvailableFontFamilyNames().
```

but there were errors converting the method call returning String array. I tried to fix this as it seemed to be simple at first, but it goes deeper. I will create new Redmine issue for this.

#28 - 04/17/2020 07:39 AM - Greg Shah

The changes are good. I've made some additional tweaks.

Eugenie: I'd like to include a list of the commonly used (in 4GL) bitmap fonts and their replacements in FWD. Can you provide such a list?

#29 - 04/17/2020 09:00 AM - Eugenie Lyzenko

Greg Shah wrote:

The changes are good. I've made some additional tweaks.

Eugenie: I'd like to include a list of the commonly used (in 4GL) bitmap fonts and their replacements in FWD. Can you provide such a list?

You mean not specific to any project, but any defaults suggested to be used from "fresh" OpenEdge installation, right?

#30 - 04/17/2020 09:10 AM - Greg Shah

Yes. When you designed the default freely licensed fonts for FWD, this included replacing the common/standard bitmap fonts. I want to document

that.

#31 - 04/17/2020 11:28 AM - Eugenie Lyzenko

Greg Shah wrote:

Yes. When you designed the default freely licensed fonts for FWD, this included replacing the common/standard bitmap fonts. I want to document that.

I think the list is:

```
MS Serif Regular or Times Roman(serife.fon) -> Liberation Mono or DejaVu Sans or not in use
MS Sans Serif Regular or Arial or Helvetica(sserife.fon) -> Liberation Sans or Wine Tahoma
FixedSys Regular (vgafix.fon) -> DejaVu Sans Bold
Courier Regular (coure.fon) -> Liberation Mono
Small Font Regular (smalle.fon) -> Liberation Mono, size=6 or not in use
```

#32 - 04/17/2020 03:44 PM - Sergey Ivanovskiy

It seems that these changes

```
=== modified file 'src/com/goldencode/p2j/ui/client/FontManager.java'
--- src/com/goldencode/p2j/ui/client/FontManager.java      2019-04-23 19:31:49 +0000
+++ src/com/goldencode/p2j/ui/client/FontManager.java      2020-04-16 22:06:00 +0000
@@ -2,7 +2,7 @@
 ** Module      : FontManager.java
 ** Abstract    : client-side font management class
 **
- ** Copyright (c) 2011-2019, Golden Code Development Corporation.
+ ** Copyright (c) 2011-2020, Golden Code Development Corporation.
 **
 ** -#- -I- --Date-- -----Description-----
 ** 001 SIY 20111115 Created initial version
@@ -52,6 +52,7 @@
 **      SBI 20180815 Changed according to GuiDriver.isFontInstalled.
 ** 033 SVL 20180913 Removed SORT_NUMBER_FONT as sort numbers are rendered using images.
 ** 034 HC  20190423 Implemented direct font control, AKA custom font support.
+ ** 035 HC  20200416 Extended direct font control to allow to use system font names.
 */

/*
@@ -1205,7 +1206,10 @@

    String fontDataKey = name.toLowerCase() + "#" + bold;
    byte[] data = wa.customFontData.get(fontDataKey);
-    if (data == null)
+
+    // proceed if the custom font data is not loaded, the font may be supported by the driver
+    GuiDriver driver = (GuiDriver) OutputManager.getDriver();
+    if (!driver.isFontInstalled(name, false))
    {
        return null;
    }
}
```

```
}
```

are not consistent with isFontInstalled usage

```
// proceed if the custom font data is not loaded, the font may be supported by the driver
GuiDriver driver = (GuiDriver) OutputManager.getDriver();
if (!driver.isFontInstalled(name, false))
{
    return null;
}
```

```
fd = new FontDetails();
fd.key = key;
fd.fontName = name;
fd.pointSize = size;
fd.style = FontStyle.createStyle(bold, italic, underline);
fd.fontDefinition = data;
```

```
// createFont will take care of caching the font
return wa.createFont(fd);
```

because of public FontDetails<?> createFont(FontDetails<?> details). If the font is not installed, then it can be loaded.

#33 - 04/17/2020 04:56 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

It seems that these changes
[...]

are not consistent with isFontInstalled usage
[...]

because of public FontDetails<?> createFont(FontDetails<?> details). If the font is not installed, then it can be loaded.

You are right, I missed a check for the custom font data. Please review 4231b revision 11467.

#34 - 04/17/2020 06:32 PM - Hynek Cihlar

Hynek Cihlar wrote:

but there were errors converting the method call returning String array. I tried to fix this as it seemed to be simple at first, but it goes deeper. I will create new Redmine issue for this.

See [#4620](#).

#35 - 04/20/2020 02:32 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

Sergey Ivanovskiy wrote:

It seems that these changes
[...]

are not consistent with isFontInstalled usage

[...]
because of public FontDetails<?> createFont(FontDetails<?> details). If the font is not installed, then it can be loaded.

You are right, I missed a check for the custom font data. Please review 4231b revision 11467.

The changes are good. I tested the button-list and calendar test cases. If fonts are changed, then UI reflects these changes. We have some functions that can return the metric size of a string from the driver, the Swing driver or the JS web driver and these metrics look calculated incorrectly in the case of font changes. I don't understand why it happens. But it is another unrelated issue. I just warn about this issue.

#36 - 04/20/2020 04:06 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

We have some functions that can return the metric size of a string from the driver, the Swing driver or the JS web driver and these metrics look calculated incorrectly in the case of font changes. I don't understand why it happens. But it is another unrelated issue. I just warn about this issue.

What is the use case? Do you have an isolated test case?

#37 - 06/20/2020 11:52 AM - Greg Shah

Task branch 4231b has been merged to trunk as revision 11347.