User Interface - Bug #2930

Feature # 1811 (Closed): implement the AJAX client driver

test IE browser support

12/12/2015 07:29 AM - Greg Shah

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Sergey Ivanovskiy	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	GUI Support for a Complex ADM2 App		
billable:	No	**case_num:**	
vendor_id:	GCD		

Description

Related issues:

Related to User Interface - Bug #2964: fix key processing for web chui in Fir...	**New**	**01/21/2016**

History

#1 - 12/12/2015 07:29 AM - Greg Shah

Please test the web client from the IE 11 browser. Report any found issues here. Resolve those issues. Anything risky can be posted here as a diff. If the changes are not risky, they can be directly checked into 1811t.

For testing, please run through the standalone GUI testcases list (the same ones we use for regression testing GUI). All testing should be done on branch 1811t.

#2 - 12/16/2015 08:21 AM - Greg Shah

Microsoft provides VMs for free to allow web developers to test their apps in various versions of the IE/Edge browsers. See:

https://dev.windows.com/en-us/microsoft-edge/tools/vms/windows/

Pick IE11 on Windows 7.

#3 - 12/16/2015 02:17 PM - Sergey Ivanovskiy

Trying to load https://192.168.1.35:7443/gui throws an invalid certificates message, it can be passed by adding https://192.168.1.35:7443/gui to trust zone. Finally I have got properly functional application except of I can't drag the virtual panel bar.

#4 - 12/16/2015 02:24 PM - Sergey Ivanovskiy

IE 10 emulation doesn't work due to Map usages, but the last Edge (11) works with Map.

#5 - 12/16/2015 04:00 PM - Sergey Ivanovskiy

- File IE_check_for_drag_event.txt added

Committed revision 10966 fixes the task bar panel dragging for IE.

#6 - 12/16/2015 04:11 PM - Sergey Ivanovskiy

For IE 11 the fonts don't work, trying to load the target font throws this exception:

```
CSS3114
"@font-face failed OpenType embedding permission check. Permission must be Installable." 
The font-face doesn't have permissions to install with the current webpage.
Get the correct permission or licenses for embedding the font.
```

Please look at
http://stackoverflow.com/questions/5587956/make-adobe-fonts-work-with-css3-font-face-in-ie9
http://www.kriesi.at/support/topic/fonts-not-loading-on-ie/
http://www.fontsquirrel.com/blog/2010/11/troubleshooting-font-face-problems
http://codecanyon.net/forums/thread/css3114-fontface-failed-opentype-embedding-permission-check-permission-must-be-installable/78963
http://carnage-melon.tom7.org/embed/

#7 - 12/21/2015 06:03 AM - Sergey Ivanovskiy

It seems that for IE it needs to convert TTF licensed fonts to EOT fonts. Please look at
https://www.kirsle.net/wizards/ttf2eot.cgi
http://www.w3.org/Submission/2008/SUBM-EOT-20080305/

#8 - 12/30/2015 09:50 PM - Greg Shah

I agree that for custom font support on IE, it looks like we will have to make some changes to support .eot filesand/or possibly some other changes to enable the fonts to load (e.g. setting the embedded install bit). Please test this out and confirm that it works. Are the changes only needed to the font files or do we need code/CSS/HTML changes too?

The other thing we need to do here is to avoid loading custom fonts if the font already exists in the target browser. In the IE case, I suspect that all of the fonts we are trying to load are already available since we are just trying to duplicate the Windows font environment on other platforms. Please investigate how we can do this and post your findings here.

#9 - 12/30/2015 10:06 PM - Greg Shah

Please see #2928-37 for details about the remote launching issues with IE.

#10 - 01/02/2016 09:44 AM - Sergey Ivanovskiy

- File p2j.eot_fonts.zip added

> I agree that for custom font support on IE, it looks like we will have to make some changes to support .eot filesand/or possibly some other changes to enable the fonts to load (e.g. setting the embedded install bit). Please test this out and confirm that it works. Are the changes only needed to the font files or do we need code/CSS/HTML changes too?

These converted fonts are fitted for IE11, but they don't work for Firefox due to this exception "downloadable font: rejected by sanitizer". Thus it requires to support EOT and TTF formats simultaneously. I have changed the font section of directory.xml to

```
        <node class="container" name="custom-fonts">
          <node class="container" name="font1">
            <node class="string" name="font">
              <node-attribute name="value" value="Tahoma"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/tahoma.eot"/>
            </node>
          </node>
          <node class="container" name="font2">
            <node class="string" name="font">
              <node-attribute name="value" value="Tahoma"/>
            </node>
            <node class="boolean" name="bold">
              <node-attribute name="value" value="TRUE"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/tahomabd.eot"/>
            </node>
          </node>
          <node class="container" name="font3">
            <node class="string" name="font">
              <node-attribute name="value" value="Courier New"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/cour.eot"/>
            </node>
          </node>
          <node class="container" name="font4">
            <node class="string" name="font">
              <node-attribute name="value" value="Courier New"/>
            </node>
            <node class="boolean" name="bold">
              <node-attribute name="value" value="TRUE"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/courbd.eot"/>
            </node>
          </node>
          <node class="container" name="font5">
            <node class="string" name="font">
              <node-attribute name="value" value="Segue UI"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/segoeui.eot"/>
            </node>
          </node>
          <node class="container" name="font6">
            <node class="string" name="font">
              <node-attribute name="value" value="Segue UI"/>
            </node>
            <node class="boolean" name="bold">
              <node-attribute name="value" value="TRUE"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/segoeuibd.eot"/>
            </node>
          </node>
          <node class="container" name="font7">
            <node class="string" name="font">
              <node-attribute name="value" value="Microsoft Sans Serif"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/micross.eot"/>
            </node>
          </node>
          <node class="container" name="font8">
            <node class="string" name="font">
              <node-attribute name="value" value="Fixedsys"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/fsex301-l2.eot"/>
            </node>
          </node>
          <node class="container" name="font9">
            <node class="string" name="font">
              <node-attribute name="value" value="MS Sans Serif"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/micross.eot"/>
            </node>
          </node>
        </node>
```

in order to check EOT fonts.

Are there another font formats fitted for all modern browsers: Firefox and Internet Explorer?

> The other thing we need to do here is to avoid loading custom fonts if the font already exists in the target browser. In the IE case, I suspect that all of the fonts we are trying to load are already available since we are just trying to duplicate the Windows font environment on other platforms. Please investigate how we can do this and post your findings here.

Ok. Planning to investigate it.

#11 - 01/02/2016 03:34 PM - Sergey Ivanovskiy

- File embed.c added

There is another method to transform ttf files found here http://carnage-melon.tom7.org/embed/ and it seems that it works for IE and Firefox simultaneously. If my understanding of this method is correct, then the program embed.c changes only one field called fstype (http://www.microsoft.com/typography/otspec/os2.htm#fst) to the 0x0000 value

```
0x0000 - Installable Embedding: No fsType bit is set. Thus fsType is zero. Fonts with this setting indicate that they may be embedded and permanently installed on the remote system by an application. The user of the remote system acquires the identical rights, obligations and licenses for that font as the original purchaser of the font, and is subject to the same end-user license agreement, copyright, design patent, and/or trademark as was the original purchaser.
```

#12 - 01/04/2016 03:49 PM - Sergey Ivanovskiy

Greg Shah wrote:

> The other thing we need to do here is to avoid loading custom fonts if the font already exists in the target browser. In the IE case, I suspect that all of the fonts we are trying to load are already available since we are just trying to duplicate the Windows font environment on other platforms. Please investigate how we can do this and post your findings here.

I have found these JS sources that can detect if the target font is installed on the system by comparing with predefined fonts:
http://www.kirupa.com/html5/detect_whether_font_is_installed.htm
https://github.com/derek1906/jFont-Checker/
https://remysharp.com/2008/07/08/how-to-detect-if-a-font-is-installed-only-using-javascript
It seems that DOJO has no such functionality. We can direct a browser to try a local font first by providing src: local('Font Name')), url(...); in the css font definition (https://developer.mozilla.org/en-US/docs/Web/CSS/%40font-face)

```
=== modified file 'src/com/goldencode/p2j/ui/client/driver/web/res/p2j.fonts.js'
--- src/com/goldencode/p2j/ui/client/driver/web/res/p2j.fonts.js    2015-10-09 19:15:11 +0000
+++ src/com/goldencode/p2j/ui/client/driver/web/res/p2j.fonts.js    2016-01-03 07:24:44 +0000
@@ -123,7 +123,7 @@
       if (b64Font != "")
       {
          var cssFont = "@font-face {  font-family: '" + name + "';";
-         cssFont = cssFont + " src: url('data:application/x-font-truetype;base64," + b64Font + "')";
+         cssFont = cssFont + " src: local('" + name + "'), url('data:application/x-font-truetype;base64," + b
64Font + "')";
          cssFont = cssFont + " format('truetype'); }";

          var newStyle = document.createElement('style');
```

#13 - 01/05/2016 09:04 AM - Sergey Ivanovskiy

Now, I have tried to resolve this issue, my server directory defines the following custom fonts

```
        <node class="container" name="custom-fonts">
          <node class="container" name="font1">
            <node class="string" name="font">
              <node-attribute name="value" value="Tahoma"/>
            </node>
            <node class="string" name="file">
```

```
              <node-attribute name="value" value="fonts/tahoma.ttf"/>
            </node>
          </node>
          <node class="container" name="font2">
            <node class="string" name="font">
              <node-attribute name="value" value="Tahoma"/>
            </node>
            <node class="boolean" name="bold">
              <node-attribute name="value" value="TRUE"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/tahomabd.ttf"/>
            </node>
          </node>
          <node class="container" name="font3">
            <node class="string" name="font">
              <node-attribute name="value" value="Courier New"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/cour.ttf"/>
            </node>
          </node>
          <node class="container" name="font4">
            <node class="string" name="font">
              <node-attribute name="value" value="Courier New"/>
            </node>
            <node class="boolean" name="bold">
              <node-attribute name="value" value="TRUE"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/courbd.ttf"/>
            </node>
          </node>
          <node class="container" name="font5">
            <node class="string" name="font">
              <node-attribute name="value" value="Segue UI"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/segoeui.ttf"/>
            </node>
          </node>
          <node class="container" name="font6">
            <node class="string" name="font">
              <node-attribute name="value" value="Segue UI"/>
            </node>
            <node class="boolean" name="bold">
              <node-attribute name="value" value="TRUE"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/segoeuibd.ttf"/>
            </node>
          </node>
          <node class="container" name="font7">
            <node class="string" name="font">
              <node-attribute name="value" value="Microsoft Sans Serif"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/micross.ttf"/>
            </node>
          </node>
          <node class="container" name="font8">
            <node class="string" name="font">
              <node-attribute name="value" value="Fixedsys"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/fsex301-l2.ttf"/>
            </node>
          </node>
          <node class="container" name="font9">
            <node class="string" name="font">
              <node-attribute name="value" value="MS Sans Serif"/>
            </node>
            <node class="string" name="file">
              <node-attribute name="value" value="fonts/micross.ttf"/>
            </node>
          </node>
        </node>
```

But the JS client (executing on Firefox Ubuntu) created 3 font-face definitions for 'MS Sans Serif', one for 'Tahoma' and one for 'Courier New'. The tested fonts checker found that 'Courier New' is only installed font from 'MS Sans Serif', 'Tahoma' and 'Courier New'. I don't understand why the same font name, 'MS Sans Serif', has been appeared 3 times.

#14 - 01/05/2016 11:51 AM - Greg Shah

> There is another method to transform ttf files found here http://carnage-melon.tom7.org/embed/ and it seems that it works for IE and Firefox simultaneously.

This seems to be a reasonable solution for now. It avoids the need to support both EOT and TTF versions of the fonts, which is good.

The only down side is that this requires modifying the font files themselves, which may be a license violation. It will completely depend on the the license for each font and we have no way of knowing right now it if will be legal.

On the other hand, we are already planning to find freely licensed alternative fonts to replace the MS fonts (see #2765). When we do that, the license should allow this change. Those fonts may not even need this change. Clearly, the MS fonts do require it.

For now this is an acceptable workaround.

> We can direct a browser to try a local font first by providing src: local('Font Name')), url(...); in the css font definition

One concern with this approach is that we would still have to download the custom font in this case. I think it is much better to avoid the download if the font already exists.

> If we don't want to send the binary fonts to the client, then we need to request the client if the target fonts are installed or not. It will make the code more complicated.

Understood. This is acceptable in order to avoid the download.

> But the JS client (executing on Firefox Ubuntu) created 3 font-face definitions for 'MS Sans Serif', one for 'Tahoma' and one for 'Courier New'. The tested fonts checker found that 'Courier New' is only installed font from 'MS Sans Serif', 'Tahoma' and 'Courier New'. I don't understand why the same font name, 'MS Sans Serif', has been appeared 3 times.

The font definitions you posted look OK.

Can you explain how you found that 'MS Sans Serif' appears 3 times? What is the JS code that showed you that result?

#15 - 01/05/2016 03:10 PM - Sergey Ivanovskiy

> > But the JS client (executing on Firefox Ubuntu) created 3 font-face definitions for 'MS Sans Serif', one for 'Tahoma' and one for 'Courier New'. The tested fonts checker found that 'Courier New' is only installed font from 'MS Sans Serif', 'Tahoma' and 'Courier New'. I don't understand why the same font name, 'MS Sans Serif', has been appeared 3 times.
>
> The font definitions you posted look OK.
>
> Can you explain how you found that 'MS Sans Serif' appears 3 times? What is the JS code that showed you that result?

I have logged in the JS console all invocations of p2j.fonts.js:createFont. For my environment the fonts are created in this order

```
bold 8pt 'sans-serif'
8pt 'MS Sans Serif'
8pt 'Courier New'
8pt 'Tahoma'
10pt 'MS Sans Serif'
bold 8pt 'MS Sans Serif'
```

and the JS fonts checker (https://remysharp.com/2008/07/08/how-to-detect-if-a-font-is-installed-only-using-javascript) detected
'sans-serif' and 'Courier New' are installed but http://www.kirupa.com/html5/detect_whether_font_is_installed.htm can detect only 'sans-serif'. Also we can use Firefox Developers tool - inspector to look at the created fonts definitions.
May be it is a bug.

#16 - 01/07/2016 05:46 AM - Sergey Ivanovskiy

I tested the changes to eliminate loading of the installed fonts and startled with the following problem.
The fonts checker detects 'MS Sans Serif' as an installed font for IE, Firefox and Google Chrome running on Window7 but the substitution font looks differently from the case if this embedded font has been downloaded from the web client (the embedded web server).

#17 - 01/07/2016 07:50 AM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

> I tested the changes to eliminate loading of the installed fonts and startled with the following problem.
> The fonts checker detects 'MS Sans Serif' as an installed font for IE, Firefox and Google Chrome running on Window7 but the substitution font looks differently from the case if this embedded font has been downloaded from the web client (the embedded web server).

The cause of the different look&feel is this code in FontTable

```
if (isBold)
{
   if (fontDefinition == null)
   {
      boldStyle = true;

      // check the non-bold definition; if it exists, then let the style use bold AND use
      // that non-bold definition
      fontDefinition = getCustomFontDefinition(fontName, false);
   }
   else
   {
      // there is a bold definition, do not re-bold the font
      boldStyle = false;
   }
}

// for bold fonts with custom definition, setting its "bold" style results in incorrect
// drawing in some drivers (i.e. Web). for this reason, when creating the style, the "bold"
// is removed, so that the font can be drawn using its look&feel as defined in the .ttf
FontStyle style = FontStyle.createStyle(boldStyle,
                                        isItalic,
                                        isUnderline);
```

One is in the title's drawing. The title font is bold 8pt Tahoma, but the bold style is removed and if we are planning to use installed fonts it should be added again.
Thus planning to prepare my changes.

#18 - 01/07/2016 03:16 PM - Sergey Ivanovskiy

Constantin, what do you think if we will move the bold embedded fonts style logic from FontTable.buildFontSpecification(String fontID, String font) to GuiWebDriver.createFont(...)?

#19 - 01/07/2016 04:13 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Constantin, what do you think if we will move the bold embedded fonts style logic from FontTable.buildFontSpecification(String fontID, String font) to GuiWebDriver.createFont(...)?

What I can say for sure is that setting bold style for a .ttf (or .eot) which has already a builtin bold effect, the text will be rendered incorrectly (there was an issue to fix the window title in the web client). Please do some experiments in both Linux and Windows OS, with what is happening if you load a custom bold .ttf font spec via CSS and then draw some text using a bold style, too (so there is a "double-bold"). Also, in Windows OS, at some point the OS may decide to resolve a i.e. "bold arial" to a .ttf file for the bold font, and use that, instead of applying bold to the normal .ttf font. I think it's a little tricky with already installed fonts, in Windows - maybe give the custom font some other unique name, so it will not collide with an already installed font in Windows? This should work, as the .ttf loaded via CSS is referenced via the name you give when the font is defined.

#20 - 01/07/2016 07:32 PM - Sergey Ivanovskiy

- File fonts1.zip added

Greg, Constantin, please review committed revision 10974. The fonts detector from https://www.samclarke.com/2013/06/javascript-is-font-available/ is used. Now the fonts are uploaded if they are not detected. The attached font1.zip contains the ttf fonts converted by embed.c (see #note-11 for the reference). I have not checked yet if we change the embedded font name to the custom name, then it helps to fix bold style issue in the title for "double bold" fonts.

#21 - 01/08/2016 09:25 AM - Greg Shah

Sergey: we can't embed this separately copyrighted code into our project. It is just something posted on a web site, so it is not a "real" project. Because of this, we can't be assured of its legitimacy AND we cannot be assured that others will be able to determine its legitimacy later (e.g. if the web site changes or goes away).

Please implement our own code for this detection, based on reading the sites that you have previously noted.

#22 - 01/08/2016 09:28 AM - Greg Shah

The other thing is that it seems like this installed font checking should be in the GuiDriver API and then the conditional installation can be done in common code (e.g. FontManager). This would also handle the case where the Swing client is running on Windows.

#23 - 01/08/2016 09:40 AM - Sergey Ivanovskiy

Greg Shah wrote:

> The other thing is that it seems like this installed font checking should be in the GuiDriver API and then the conditional installation can be done in common code (e.g. FontManager). This would also handle the case where the Swing client is running on Windows.

Greg, please clarify this requirement, because at first we have looked at the web case and the swing client case has not been in focus.

#24 - 01/08/2016 10:01 AM - Sergey Ivanovskiy

For the Swing client installed fonts can be enumerated and are configured by the java environment. This case is more simpler than the web case.

#25 - 01/08/2016 10:03 AM - Greg Shah

> at first we have looked at the web case and the swing client case has not been in focus

I understand you have not considered it, but I was always intending for this to be implemented in common code using a driver API for testing if the font is installed. I should have explained this more clearly before.

#26 - 01/08/2016 10:05 AM - Sergey Ivanovskiy

> I understand you have not considered it, but I was always intending for this to be implemented in common code using a driver API for testing if the font is installed. I should have explained this more clearly before.

Ok, I see, thank you. Planning to resolve this case.

#27 - 01/10/2016 08:23 AM - Sergey Ivanovskiy

- File installed_fonts_1.txt added

Greg, please review the committed revision 10976, isFontInstalled(String fontName) is moved to GuiDriver and the fonts check is moved to FontManager.

#28 - 01/10/2016 10:00 AM - Sergey Ivanovskiy

The committed revision 10977 is done in order to remove the author reference since it can be used here.

#29 - 01/11/2016 03:31 PM - Greg Shah

Code Review Task Branch 1811t Revision 10976

SwingGuiDriver.isFontInstalled() needs javadoc. Otherwise the changes are good.

#30 - 01/11/2016 03:32 PM - Greg Shah

Code Review Task Branch 1811t Revision 10977

The FontChecker code is still too much like the original copied code. We cannot use the code as it is. It must be your own version of the same idea. It must be written "from scratch".

#31 - 01/12/2016 04:12 PM - Sergey Ivanovskiy

Greg Shah wrote:

> Code Review Task Branch 1811t Revision 10977
>
> The FontChecker code is still too much like the original copied code. We cannot use the code as it is. It must be your own version of the same idea. It must be written "from scratch".

The FontChecker code is standard and it is difficult to optimize. These two https://www.samclarke.com/2013/06/javascript-is-font-available/ and https://github.com/derek1906/jFont-Checker/blob/master/jfont-checker.js have the common method, but https://www.samclarke.com/2013/06/javascript-is-font-available/ has tested a given font with
monospace, serif and sans-serif. The last ones are substituted by browser available physical fonts. But https://github.com/derek1906/jFont-Checker/blob/master/jfont-checker.js tests fonts against a monospace font. I have no idea how to change it because all it steps are almost optimal and simple. May be do change names?

#32 - 01/13/2016 02:32 AM - Sergey Ivanovskiy

Greg, please review the committed 10980.

#33 - 01/13/2016 09:20 AM - Greg Shah

> May be do change names?

Changing names is useful, but by itself it is not enough.

However, you added caching and used a different test function that integrates with the caching implementation. That combined with your previous changes is almost enough changes that the code is unique (and thus is not a copy of the original code).

I think that the last change needed is a different way to generate the text. Do not use Array(100).join('wi'). Pick other text that is uniquely yours. I think the key is to have enough variability of character widths such that the differences across fonts will always be encountered.

#34 - 01/13/2016 02:35 PM - Sergey Ivanovskiy

Fixed FontsChecker sample text committed rev 10982.

#35 - 01/14/2016 10:10 AM - Greg Shah

Code Review Task Branch 1811t Revision 10982

The changes are fine.

Let me know when you believe your testing is complete.

#36 - 01/25/2016 04:06 PM - Greg Shah

From Sergey:

Propose to close the tasks 2929 and 2930 and to support only the required keys combinations for the converted application logic. In general all keys are supported.

#37 - 01/25/2016 04:07 PM - Greg Shah

- % Done changed from 0 to 100

- Status changed from New to Closed

Please add any additional found key exclusions to #2675.

#38 - 11/16/2016 12:12 PM - Greg Shah

- Target version changed from Milestone 12 to GUI Support for a Complex ADM2 App

Files

IE_check_for_drag_event.txt	3.21 KB	12/16/2015	Sergey Ivanovskiy
p2j.eot_fonts.zip	2.59 MB	01/02/2016	Sergey Ivanovskiy
embed.c	2.17 KB	01/02/2016	Sergey Ivanovskiy
fonts1.zip	2.59 MB	01/08/2016	Sergey Ivanovskiy
installed_fonts_1.txt	5.76 KB	01/10/2016	Sergey Ivanovskiy