

Base Language - Support #1700

test the startup, basic processing and shutdown of the P2J server and P2J client on Windows and log all found issues

10/24/2012 11:10 AM - Greg Shah

Status:	Closed	Start date:	10/24/2012
Priority:	Normal	Due date:	
Assignee:	Eugenie Lyzenko	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	Runtime Support for Server Features	case_num:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 10/24/2012 11:16 AM - Greg Shah

- Status changed from New to WIP
- Assignee set to Eugenie Lyzenko

Use the p2j/testcases/simple/ environment to test the server and client on Windows. For now, DON'T test any of the Progress 4GL file processing (for example, OUTPUT STREAM mystream TO some\filename.txt). Just pick a .p that will allow non-filesystem processing to be run.

DON'T try to use the Charva client. For now, just use the Swing client which is the default for the simple/client/client.sh, since this should already be working pretty well on Windows.

Identify any problems and log them in Redmine. If they block your testing and you can easily fix them, then go ahead and propose changes.

#2 - 10/25/2012 03:52 PM - Eugenie Lyzenko

- File LucidaSansTypewr.jpg added
- File monospaced.jpg added
- File monospaced_Linux.jpg added

The first issue/difference has been found running swing client in Windows. Look at the screenshots attached. If we use monospaced font for client the ChUI screen looks corrupted and may be the selected font is not really monospaced(monospaced.jpg). Compare to the Linux counterpart(monospaced_linux.jpg). To have the same screen in Windows I had to change the font to explicit "Lucida Sans Typewriter" and this makes the Windows-Linux screens equivalent(LucidaSansTypewr.jpg).

#3 - 10/26/2012 06:35 PM - Eugenie Lyzenko

- File client.cmd added
- File server.cmd added
- File directory.xml.win added

Currently the Swing client starting does not have any obvious issues except font rendering issue already found. Additional investigation should be performed to find any possible constraints here. There are two new files to start server/client under Windows: server.cmd and client.cmd. Just go to the p2j\testcases\simple\server or p2j\testcases\simple\client directories and start the commands. Not all Linux command file logic can be migrated in

Windows. Particularly instead of PID stamp in client log file the current date_time stamp has been included to form the filename.

Also the directory.xml file should be modified to take into account Windows specific for path|file separators and prospath value. All new files are uploaded here.

#4 - 10/31/2012 01:31 PM - Greg Shah

- Target version set to Milestone 7

#5 - 11/09/2012 04:03 PM - Eugenie Lyzenko

- File server.cmd added

- File client.cmd added

The new tests does not have the issues except one that related to the Swing screen representation. The basic functionality is working OK. The editing fields, selections text drawing.

However the trying to set up monospaced fonts in Windows does not have the correct screen results in Windows. The debugging shows the monospaced font is mapping to the system "Courier New", while in Linux it is "Lucida Sans Typewriter". Except the difference in cell height the selected monospaced font in Windows is not really monospaced. This is strange and looks like a bug in JVM implementation in Windows or Windows font rendering subsystem itself. The symptoms are the line width is dependent on the contents of the line especially when the line has many " " characters or selection bar.

So the solution is to specify the monospaced font deployed with the Java SE for Windows. The "Lucida Sans Typewriter" works fine as option. The sample files to start server and client are uploaded here.

#6 - 11/12/2012 11:43 AM - Greg Shah

Package the changes as a proper update zip. These do not affect anything in the conversion or runtime, so we will just apply them to staging and check them into Bazaar without further testing.

After that, please examine the com/goldencode/p2j/ui/client/swing/SwingHelper class. In there you will find isMonospaced() and getMonoFontList(). I suspect the problem is in our test for whether something is a monospaced font. Please debug through there and see if you can identify the root cause of the font problem.

#7 - 11/12/2012 01:34 PM - Eugenie Lyzenko

- File evl_upd20121112a.zip added

Package the changes as a proper update zip. These do not affect anything in the conversion or runtime, so we will just apply them to staging and check them into Bazaar without further testing.

The update has been uploaded here. It contains two cmd script files, one for server and one for client. Should I apply them to the staging and check into bzt?

Usage instruction:

1. It is not required to adjust server's directory file. Only change the sample execution name as it is required in Linux.

2. Convert sample using conv2.cmd from testcases/uast subdir.
3. cd p2j/testcases/simple/server, run server.cmd
4. cd p2j/testcases/simple/client, run client.cmd

After that, please examine the com/goldencode/p2j/ui/client/swing/SwingHelper class. In there you will find isMonospaced() and getMonoFontList(). I suspect the problem is in our test for whether something is a monospaced font. Please debug through there and see if you can identify the root cause of the font problem.

OK.

#8 - 11/12/2012 06:01 PM - Eugenie Lyzenko

- File deleted (evl_upd20121112a.zip)

#9 - 11/12/2012 06:03 PM - Eugenie Lyzenko

- File evl_upd20121112a.zip added

The new version with commented out the dead command has been uploaded.

#10 - 11/13/2012 11:56 AM - Greg Shah

The only change needed is to re-zip the update without the p2j/ prefix. Zip everything from inside the p2j/ directory and it will be good to go.

Attach that final zip to this task. Then you can check it in and apply it to staging.

To clarify on the directory changes needed: it seems that the required changes are in the file-system section (propath, file-separator, path-separator).

#11 - 11/13/2012 01:29 PM - Eugenie Lyzenko

- File evl_upd20121113a.zip added

The update has been repackaged. Also it includes the directory.xml.win and directory.xml.linux files for both systems. The differences are caused by special values for file-separator, path-separator and propath values. Just copy the respective file to directory.xml one depending on the system to run under.

The changes will now be checked in and applied to staging. No regression testing required.

#12 - 11/13/2012 01:55 PM - Eugenie Lyzenko

Applied to staging and checked in Bazaar.

#13 - 11/16/2012 07:35 AM - Eugenie Lyzenko

- File mono_vs_courier.jpg added

The debugging results:

1. The `SwingHelper.getMonoFontList()` and `SwingHelper.isMonospaced()` are seems to work fine. The first method is called only if we invoke settings dialog to manually change the current font. But when we start the sample Swing application the `SwingHelper.getMonoFontList()` is not called. We just request the Logical monospaced font.

2. Take a look at the new picture attached - `mono_vs_courier.jpg`. I have combined output from two layouts, one with Monospaced font(with white lines) and parts from Courier New layout. I think we can conclude the font is really the same for both cases. So Windows maps Monospaced to Courier New. And the width of the letters and other chars are the same(7). But as you can see the width of the " " character is very different. The cursor shape(top left corner) is much bigger in Monospaced case. And the height of the monospaced logical font is more than Courier New(16 for mono, 13 for Courier).

3. The other interesting thing is selecting the Monospaced font in other applications(jEdit for example) causes the same issues with text - the shapes looks like monospaced but the " " char is different.

I have experimented with following approach:

- get the list of all mono font by `The SwingHelper.getMonoFontList()`
- select the first available font from returning array

In Windows this works OK and uses Courier New. In Linux this does not work. So we can not use this approach.

But I guess to provide good looking on Windows we need to select Physical font instead of the Logical. For example, call `SwingHelper.getMonoFontList()` and select the best suitable from the returned list. Continue debugging.

#14 - 11/16/2012 04:31 PM - Eugenie Lyzenko

More investigation.

Looks like the reason for screen corruption in Windows monospaced has been found. The CHUI output is a combination of two types of chars: one - letters and " " and another - so called "Box Drawing" chars(frame border chars starting with `\u2500`). And on the other hand the Java logical font can be composite meaning one logical font can be mapped to the several physical ones to cover the full Unicode range if single physical font can not render all required characters. In our case the "Box Drawing" characters are really wider than basic Latin charset(0x00 - 0x7F).

So we need monospaced font is able to draw the both letters and frame border chars simultaneously. In this case all required characters will have the same size. I would offer to use the helper function like below before first usage of the ChUI client:

Class `SwingHelper`:

```
public static synchronized String getMonoFont(Component comp, String defaultFont) {
    String[] all = getMonoFontList(comp);
    for (int i = 0; i < all.length; i++) {
        Font font = Font.decode(all[i]);
        if (font.canDisplay("\u2500")) {
            FontMetrics fm = comp.getFontMetrics(font);
            if (fm.charWidth('i') == fm.charWidth("\u2500"))
                return all[i];
        }
    }
    return defaultFont;
}
```

The method returns unchanged font name if we can not find required font and this is the last resort to trust the Java logical font selection. Yes, we will use here the physical font but it is platform neutral. For every OS the choice will be different. It is tested in both Windows and Linux and works fine. In my Windows it selects real "Courier New". In my Linux box it selects DejaVu Sans Mono. What do you think about this solution?

#15 - 11/16/2012 05:21 PM - Greg Shah

OK, this is close. But it seems that the real problem is just with our `isMonospaced()` implementation. In other words, you have found some flaws with our testing of a font to see if it is really monospaced. So, let's change the `getMonoFontList()` to go from this:

```
Font font = Font.decode(all[i]);

if (isMonospaced(comp.getFontMetrics(font)))
{
    array.add(all[i]);
}
```

to this:

```
Font font = Font.decode(all[i]);

if (isMonospaced(font, comp.getFontMetrics(font)))
{
    array.add(all[i]);
}
```

Then change `isMonospaced()` to this:

```
public static boolean isMonospaced(Font font, FontMetrics fm)
{
    if (!font.canDisplay('\u2500'))
        return false;

    // check if this font is monospaced, this is a bit of a hack but should
    // be reliable
    int msize = fm.charWidth('m');
    int isize = fm.charWidth('i');
    int osize = fm.charWidth('1');
    int nsize = fm.charWidth('9');
    int dsize = fm.charWidth('\u2500');

    return (msize == isize && msize == nsize && msize == osize && osize == dsize);
}
```

Please review and confirm that it has the same result as your approach. The idea is to exclude any invalid fonts from the master list. Then anything from that list can be chosen safely.

#16 - 11/16/2012 06:25 PM - Eugenie Lyzenko

Your change is not enough to get the ChUI client working. Consider the constructor of the ChuiSimulator class. The getMonoFontList() is not called in the first displaying of the client. It calls configure(options) which in turns does:

```
...
    if (!SwingHelper.isMonospaced(font, fm))
    {
        throw new IllegalArgumentException("Font face '" +
            options.fontname +
            "' must be fixed width.");
    }
...

```

And this just throws exception and stops loading the client. The getMonoFontList() is called only if we activate the settings panel. We need to modify the ChuiSimulator constructor from

```
public ChuiSimulator(Frame frame,
                    ContainerOps container,
                    TerminalOptions options)
{
    configure(options);
}
...

```

to something like:

```
public ChuiSimulator(Frame frame,
                    ContainerOps container,
                    TerminalOptions options)
{
    if (options.fontname.equalsIgnoreCase("monospaced"))
        options.fontname = SwingHelper.getMonoFontList(this)[0];
    configure(options);
}
...

```

or:

```
public ChuiSimulator(Frame frame,
                    ContainerOps container,
                    TerminalOptions options)
{
    if (options.fontname.equalsIgnoreCase("monospaced"))
        options.fontname = SwingHelper.getMonoFont(this, options.fontname);
    configure(options);
}
...

```

where SwingHelper.getMonoFont:

```
public static synchronized String getMonoFont(Component comp, String defaultFont)
{
    String[] all = getMonoFontList(comp);
    if (all.length > 0)
        return all[0];
    else
        return defaultFont;
}

```

This will do the same as my previous approach.

#17 - 11/19/2012 07:42 AM - Greg Shah

Go with your 2nd approach:

```
public ChuiSimulator(Frame frame,
                    ContainerOps container,
                    TerminalOptions options)
{
    if(options.fontname.equalsIgnoreCase("monospaced"))
        options.fontname = SwingHelper.getMonoFont(this, options.fontname);
    configure(options);
    ...
}
```

Except, rename getMonoFont() to getMonoFontName() which is more descriptive.

#18 - 11/19/2012 11:22 AM - Eugenie Lyzenko

OK. What about changes testing and uploading?

The code involved in the change does not affect the usual client behavior and regression tests. Do we need to run harness for this change anyway?

Should I commit this change to the P2J bzt project? Or for now we hold this change for future releases?

#19 - 11/19/2012 11:46 AM - Greg Shah

Prepare the update zip file and attach it to this issue's history. I will then do a final code review and let you know.

#20 - 11/19/2012 03:27 PM - Eugenie Lyzenko

- File *evl_upd20121119a.zip* added

The modified versions of the ChuiSimulator.java and SwingHelper.java have been uploaded for review.

#21 - 11/19/2012 05:16 PM - Greg Shah

You can check in the change to bzt and apply it to staging.

#22 - 11/19/2012 07:32 PM - Eugenie Lyzenko

Checked in p2j bzt and applied to staging.

Need to run ant jar in staging p2j to take effect.

#23 - 11/20/2012 07:53 AM - Greg Shah

I rebuilt staging. Please distribute the update via email.

#24 - 11/20/2012 02:17 PM - Greg Shah

- % Done changed from 0 to 100
- Status changed from WIP to Closed

#25 - 11/16/2016 11:43 AM - Greg Shah

- Target version changed from Milestone 7 to Runtime Support for Server Features

#26 - 11/22/2016 10:02 AM - Greg Shah

I am removing the directory example because it contains some confidential information. The portion that was important is this (in the /server/standard/runtime/ section):

```
<node class="container" name="default">
  <node class="container" name="file-system">
    <node class="string" name="pkgroot">
      <node-attribute name="value" value="com.goldencode.testcases"/>
    </node>
    <node class="string" name="propath">
      <node-attribute name="value" value="."/>
    </node>
    <node class="string" name="path-separator">
      <node-attribute name="value" value=";"/>
    </node>
    <node class="string" name="file-separator">
      <node-attribute name="value" value="\"/>
    </node>
    <node class="boolean" name="case-sensitive">
      <node-attribute name="value" value="TRUE"/>
    </node>
  </node>
</node>
```

#27 - 11/22/2016 10:02 AM - Greg Shah

- File deleted (directory.xml.win)

Files

monospaced.jpg	37.3 KB	10/25/2012	Eugenie Lyzenko
monospaced_Linux.jpg	18.9 KB	10/25/2012	Eugenie Lyzenko
LucidaSansTypewr.jpg	20.9 KB	10/25/2012	Eugenie Lyzenko
server.cmd	2.21 KB	10/26/2012	Eugenie Lyzenko
client.cmd	3.2 KB	10/26/2012	Eugenie Lyzenko
server.cmd	2.21 KB	11/09/2012	Eugenie Lyzenko
client.cmd	3.25 KB	11/09/2012	Eugenie Lyzenko

evl_upd20121112a.zip	3.63 KB	11/12/2012	Eugenie Lyzenko
evl_upd20121113a.zip	12.3 KB	11/13/2012	Eugenie Lyzenko
mono_vs_courier.jpg	63.8 KB	11/16/2012	Eugenie Lyzenko
evl_upd20121119a.zip	13.6 KB	11/19/2012	Eugenie Lyzenko