User Interface - Bug #3066

The web window can't be resized to widen its size

04/14/2016 12:01 PM - Sergey Ivanovskiy

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Sergey Ivanovskiy	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Cleanup and Stabilization for GUI		
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 04/14/2016 12:04 PM - Sergey Ivanovskiy

For the web client an application window can be squeezed but can't be resized to widen its width and height.

#2 - 04/15/2016 03:07 AM - Sergey Ivanovskiy

I tested ./demo/simple_windows.p with the web client. WindowGuiImpl on afterConfigUpdate(WindowConfig beforeUpdate)

```
      if (beforeUpdate.minHeightPixels != config.minHeightPixels ||
          beforeUpdate.minWidthPixels  != config.minWidthPixels  ||
          beforeUpdate.maxHeightPixels != config.maxHeightPixels ||
          beforeUpdate.maxWidthPixels  != config.maxWidthPixels  ||
          beforeUpdate.resize != config.resize)
      {
         gd.selectWindow(wid);
         gd.setResizeableWindow(config.resize,
                                config.minWidthPixels,
                                config.minHeightPixels,
                                config.maxWidthPixels,
                                config.maxHeightPixels);
         gd.releaseWindow();
      }
```

produced these logs

```
SET_RESIZEABLE_WINDOW: id=1 minw=0 minh=0 maxw=400 maxh=462
SET_RESIZEABLE_WINDOW: id=1 minw=50 minh=0 maxw=400 maxh=462
SET_RESIZEABLE_WINDOW: id=1 minw=50 minh=210 maxw=400 maxh=462
SET_RESIZEABLE_WINDOW: id=23 minw=50 minh=0 maxw=0 maxh=0
SET_RESIZEABLE_WINDOW: id=23 minw=50 minh=210 maxw=0 maxh=0
```

According to this log an application window isn't resized to widen its size due to the incorrect maximal width. The actual window has the width that is greater than the provided maximal value. The web client takes into account the maximal width and height assigned to the resized window.

#3 - 04/15/2016 03:41 AM - Sergey Ivanovskiy

Committed revision 11008/11009 fixed p2j.logger.log() and history entries.

#4 - 04/15/2016 11:14 AM - Sergey Ivanovskiy

Greg, could you help? The Swing client doesn't implement setResizeableWindow, but the Web client uses it. The provided values look strange and are less than actual width and height. What do you advise to check?

#5 - 04/15/2016 01:22 PM - Greg Shah

I think you need to set breakpoints in WindowWidget.setMaxWidthPixelsWorker() and WindowWidget.updateMinMaxSize(), which is where the max*Pixels gets assigned.

Hynek: Any other thoughts?

#6 - 04/15/2016 01:30 PM - Hynek Cihlar

Greg Shah wrote:

> I think you need to set breakpoints in WindowWidget.setMaxWidthPixelsWorker() and WindowWidget.updateMinMaxSize(), which is where the max*Pixels gets assigned.
>
> Hynek: Any other thoughts?

Sergey, you may also set brakepoints in SyncCoordinatesAspect.afterSetMax* methods, make sure you debug both the server and the client.

#7 - 04/15/2016 07:44 PM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Greg Shah wrote:
>
> > I think you need to set breakpoints in WindowWidget.setMaxWidthPixelsWorker() and WindowWidget.updateMinMaxSize(), which is where the max*Pixels gets assigned.
> >
> > Hynek: Any other thoughts?
>
> Sergey, you may also set brakepoints in SyncCoordinatesAspect.afterSetMax* methods, make sure you debug both the server and the client.

Greg and Hynek, thank you, but the questions will be more than the answers. Why WindowWidget invokes pushScreenDefinition() only for this update methods:

```
setMessageAreaFont(int64)
setStatusAreaFont(int64)
setTopOnly(boolean)
setVisible(boolean)
```

and for the others update methods it uses LogicalTerminal.pushWindow() or doesn't send updates to the client. For example these methods setMaxWidthPixelsWorker, setMaxHeightPixelsWorker and updateMinMaxSize() are silent for the client. The default window sends its config from LogicalTerminal.init() and SyncCoordinatesAspect.afterSetMaxWidthPixels(WindowConfig cfg) is invoked, but for an ordinal window it isn't invoked.

#8 - 04/15/2016 07:51 PM - Sergey Ivanovskiy

It is clear now because this method BaseEntity<T extends BaseConfig>.setSizePixels(Integer value, SizeAttribute attr) uses setMaxWidthPixelsWorker, setMaxHeightPixelsWorker and sends updates to the client.

#9 - 04/16/2016 04:46 AM - Sergey Ivanovskiy

I tested ./demo/simple_windos.p. If my understanding is correct, then on the server side the default window configuration is fixed by invoking WindowWidget.public void afterConfigUpdate(WindowConfig beforeUpdate) on LogicTerminal.init() and for the second dynamic window this method isn't invoked on the server side and the dynamic window configuration has invalid marker values that must be replaced by the valid window width and height values. Thus I think that ConfigManager.syncConfigChanges(Runnable) must be called on the server side for each WindowWidget update methods. Now we only push window definitions to the client but doesn't notify the server itself. What do you think my proposal is correct?

#10 - 04/17/2016 10:53 AM - Greg Shah

> What do you think my proposal is correct?

This is really a question for Hynek.

Constantin: did any of your reduction in push definition/sync processing affect this area?

#11 - 04/17/2016 08:29 PM - Sergey Ivanovskiy

- File 3066_3.txt added

We can fix the unset maxHeightChars and maxWidthChars on the client side using WindowManager.setInitialDimension, but these values aren't updated if heightChars and widthChars are changed by business logic. The conversion code of the considered test (./demo/simple_windows.p) doesn't set width and height directly. minHeightChars and minWidthChars are set only

```
public void body()
{
   f1Frame.openScope();
   f2Frame.openScope();
   defaultWindow().unwrapWindow().setMinWidthChars(new decimal(new integer(10)));
   defaultWindow().unwrapWindow().setMinHeightChars(new decimal(new integer(10)));
   DynamicWidgetFactory.createWindow(h);
   h.unwrapWidget().setTitle(new character("second window"));
   h.unwrapWindow().setMinWidthChars(new decimal(new integer(10)));
   h.unwrapWindow().setMinHeightChars(new decimal(new integer(10)));
   i.assign(12345);
   ch.assign("abcdef");
```

```
   FrameElement[] elementList0 = new FrameElement[]
   {
      new Element(i, f2Frame.widgeti()),
      new Element(ch, f2Frame.widgetCh())
```

```
            };

            f2Frame.display(elementList0, h);

            FrameElement[] elementList1 = new FrameElement[]
            {
               new Element(i, f1Frame.widgeti()),
               new Element(ch, f1Frame.widgetCh()),
               new WidgetElement(f1Frame.widgetBtn())
            };

            f1Frame.update(elementList1);
         }
```

#12 - 04/17/2016 08:37 PM - Sergey Ivanovskiy

The min/maxWidthChars and min/maxHeightChars logic is unclear. These comments WindowWidget.updateMinMaxSize() are confusing:

```
      // when max attribute is never set, its value equals to the width/height
      // (it grows and shrinks with the values of width/height).
      // When set however, the max value grows with width/height but doesn't
      // shrink.
      if (config.maxWidthPixels != BaseConfig.INV_COORD &&
          config.maxWidthPixels < config.widthPixels)
      {
         config.maxWidthPixels = config.widthPixels;
      }

      if (config.maxWidthPixels != BaseConfig.INV_COORD &&
          config.maxHeightPixels < config.heightPixels)
      {
         config.maxHeightPixels = config.heightPixels;
      }
```

#13 - 04/18/2016 02:52 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

Hynek Cihlar wrote:

Greg Shah wrote:

I think you need to set breakpoints in WindowWidget.setMaxWidthPixelsWorker() and WindowWidget.updateMinMaxSize(), which is where the max*Pixels gets assigned.

Hynek: Any other thoughts?

Sergey, you may also set brakepoints in SyncCoordinatesAspect.afterSetMax* methods, make sure you debug both the server and the client.

Greg and Hynek, thank you, but the questions will be more than the answers. Why WindowWidget invokes pushScreenDefinition() only for this update methods:
[...]
and for the others update methods it uses LogicalTerminal.pushWindow()

LT.pushScreenDefinition() ends up calling LT.pushWindow() for WindowWidget, check the method's body.

or doesn't send updates to the client. For example these methods setMaxWidthPixelsWorker, setMaxHeightPixelsWorker and updateMinMaxSize() are silent for the client.

These methods themselves do not send updates but the methods that call these methods do. Check where these methods get used.

The default window sends its config from LogicalTerminal.init() and SyncCoordinatesAspect.afterSetMaxWidthPixels(WindowConfig cfg) is invoked, but for an ordinal window it isn't invoked.

Probably because your test doesn't call MAX-WIDTH-PIXELS but MAX-WIDTH-CHARS? Put a break point to SCA.afterSetMaxWidthChars(), too.

#14 - 04/18/2016 03:01 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> We can fix the unset maxHeightChars and maxWidthChars on the client side using

The max values for the non-default window are assigned somewhere.

```
WindowManager.setInitialDimension, but these values aren't updated if heightChars and widthChars are changed b
y business logic. The conversion code of the considered test (./demo/simple_windows.p) doesn't set width and h
eight directly. minHeightChars and minWidthChars are set only
```

> [...]

The max attributes seem to be zeroed in WindowWidget:

```
   public void afterConfigUpdate(WindowConfig beforeUpdate)
   {
      super.afterConfigUpdate(beforeUpdate);

      // when window is realized
      if (config.realized && !beforeUpdate.realized)
      {
         ...

         // fix max size
         if (config.maxHeightChars == BaseConfig.INV_COORD)
            config.maxHeightChars = config.heightChars;

         if (config.maxWidthChars == BaseConfig.INV_COORD)
            config.maxWidthChars = config.widthChars;
      }
   }
```

The condition above is satisfied only for non-default windows and because config.heightChars and config.widthChars are never assigned they contain zero values. Instead of config.heightChars (the value assigned on the server by 4GL code), we should use the actual (effective) window size here.

#15 - 04/18/2016 03:24 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Sergey Ivanovskiy wrote:
>
> > We can fix the unset maxHeightChars and maxWidthChars on the client side using
>
> The max values for the non-default window are assigned somewhere.
>
> WindowManager.setInitialDimension, but these values aren't updated if heightChars and widthChars are changed by business logic. The conversion code of the considered test (./demo/simple_windows.p) doesn't set width and height directly. minHeightChars and minWidthChars are set only
>
> > [...]
>
> The max attributes seem to be zeroed in WindowWidget:
> [...]

This method has been called only for the default window from LogicTerminal.init():

```
         // LT can be used during server start too (i.e. by MetadataManager.populateDatabase)
         // so ensure we have a proper client session
         ConfigManager mgr = ConfigManager.getInstance();

         if (mgr.getActiveConfig(defaultWindow.config().id) == null)
         {
            // TODO: when the context is reset (in an appserver agent), what happens with the
            // client-side? do the widgets gets destroyed too, and the window's configuration
            // reinitialized? (i.e. it needs to be pushed by the client-side again?).

            // register only once
            mgr.addWidgetConfig(defaultWindow.config());

            ConfigSyncManager.markScopeStart();
            try
            {
               // the window was registered, activate the config updates
               client.activateConfigUpdates();
               defaultWindow.config().realized = true;
            }
```

WindowWidget.afterConfigUpdate(beforeUpdate) is called from this method ConfigSyncManager.markScopeEnd(), but for the dynamic window created by ./demo/simple_windows.p
WindowWidget.afterConfigUpdate(beforeUpdate) is never called.

```
            finally
            {
               ConfigSyncManager.markScopeEnd();
            }
         }
```

> The condition above is satisfied only for non-default windows and because config.heightChars and config.widthChars are never assigned they contain zero values. Instead of config.heightChars (the value assigned on the server by 4GL code), we should use the actual (effective) window size here.

Do you mean to use the actual(effective) window size for the client side in this method WindowManager.setInitialDimension(...) to fix unset values?

#16 - 04/18/2016 04:05 AM - Hynek Cihlar

I just checked the web client with demo/simple_windows.p setting max attributes or not and I can resize the windows OK. What are the steps to reproduce this issue?

#17 - 04/18/2016 04:10 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> I just checked the web client with demo/simple_windows.p setting max attributes or not and I can resize the windows OK. What are the steps to reproduce this issue?

It is possible to decrease the size for the second dynamic window but it is not possible to restore the size after the window size has been decreased by mouse resizing.

#18 - 04/18/2016 04:12 AM - Sergey Ivanovskiy

The issue is the max width and height are unset on the server side and on the client side.

#19 - 04/18/2016 04:14 AM - Sergey Ivanovskiy

Is it true that for you this simple example works properly?

#20 - 04/18/2016 04:21 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Is it true that for you this simple example works properly?

Yes, for me simple_windows.p seems to work correctly on Web client. But there may be some steps I am missing. Pls post your exact steps.

#21 - 04/18/2016 04:22 AM - Hynek Cihlar

Hynek Cihlar wrote:

> Sergey Ivanovskiy wrote:
>
> > Is it true that for you this simple example works properly?
>
> Yes, for me simple_windows.p seems to work correctly on Web client. But there may be some steps I am missing. Pls post your exact steps.

Nevermind, I can see the problem now.

#22 - 04/18/2016 04:25 AM - Sergey Ivanovskiy

Ok, :) but the Swing client could have the same problem if it would implement the resizing bounds. It is not the web client issue. We can work with the Swing client because debugging this client and server pair is simpler than the web client and the server.

#23 - 04/18/2016 04:40 AM - Hynek Cihlar

To resolve the issue, use ConfigHelper.getMaxWidthPixels() and related methods in the call to GuiDriver.setResizeableWindow().

#24 - 04/18/2016 05:02 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> To resolve the issue, use ConfigHelper.getMaxWidthPixels() and related methods in the call to GuiDriver.setResizeableWindow().

Thanks, it will help to fix the values on the client side, but the window config leaves unchanged. Is it correct?

#25 - 04/18/2016 06:00 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Hynek Cihlar wrote:
>
> > To resolve the issue, use ConfigHelper.getMaxWidthPixels() and related methods in the call to GuiDriver.setResizeableWindow().
>
> Thanks, it will help to fix the values on the client side, but the window config leaves unchanged. Is it correct?

This is correct, the config value reflects the server-assigned value, but not the effective size.

#26 - 04/18/2016 07:38 AM - Sergey Ivanovskiy

- File 3066_4.txt added

Committed revision 11010. Please review. The weak place is that don't know if it is possible that the actual window width can be greater than the maximal width assigned by the business logic.

#27 - 04/18/2016 07:44 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> The weak place is that don't know if it is possible that the actual window width can be greater than the maximal width assigned by the business logic.

The actual width/height can be greater than max width/max height if assigned by business logic, but not when resized from user input.

#28 - 04/18/2016 07:57 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Committed revision 11010. Please review.

What branch?

#29 - 04/18/2016 08:48 AM - Sergey Ivanovskiy

Sorry, the committed rev. 11010 is in 1811u.

#30 - 04/18/2016 08:59 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Sorry, the committed rev. 11010 is in 1811u.

The changes are OK.

Let me just suggest to change Math.max(ConfigHelper.getMaxWidthPixels(wcfg), size.width) (and height) to simple ConfigHelper.getMaxWidthPixels(wcfg). This way we will align to the size-limit logic in WindowLayout.doLayout() and, in case the max values are calculated incorrectly, we will have better chance to get consistent behavior with Swing GUI.

#31 - 04/18/2016 09:07 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Sergey Ivanovskiy wrote:
>
> > Sorry, the committed rev. 11010 is in 1811u.

> The changes are OK.
>
> Let me just suggest to change Math.max(ConfigHelper.getMaxWidthPixels(wcfg), size.width) (and height) to simple ConfigHelper.getMaxWidthPixels(wcfg). This way we will align to the size-limit logic in WindowLayout.doLayout() and, in case the max values are calculated incorrectly, we will have better chance to get consistent behavior with Swing GUI.

I added this check because the configuration size was less than the actual size. It impacts how these windows are resized. Thus if we don't fix it, then we get that the resizing window can't reach its initial actual size.

#32 - 04/18/2016 01:48 PM - Greg Shah

Hynek: are you OK with me closing this task?

#33 - 04/18/2016 02:17 PM - Hynek Cihlar

Greg Shah wrote:

> Hynek: are you OK with me closing this task?

No more objections here.

#34 - 04/18/2016 02:17 PM - Greg Shah

- Status changed from New to Closed

- % Done changed from 0 to 100

- Target version set to Milestone 16

#35 - 11/16/2016 12:22 PM - Greg Shah

- Target version changed from Milestone 16 to Cleanup and Stabilization for GUI

Files

3066_3.txt	589 Bytes	04/18/2016	Sergey Ivanovskiy
3066_4.txt	2.02 KB	04/18/2016	Sergey Ivanovskiy