User Interface - Feature #3218

enhance the embedded mode web GUI to implement a good default approach for common cases

12/07/2016 03:19 PM - Greg Shah

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Constantin Asofiei	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD

Description

Related issues:

Related to User Interface - Feature #3258: improve javascript upcalls from th...	**New**
Related to User Interface - Feature #3259: add useful web components that can...	**New**
Related to User Interface - Feature #3260: auto-conversion and enablement for...	**Closed**
Related to User Interface - Feature #3261: enhanced browse that can optionall...	**Closed**

History

#1 - 12/07/2016 03:27 PM - Greg Shah

It is time to implement common-sense choices about how to display various windows in embedded mode. We can refine it later when we get feedback from a customer, but for now I want to make sure that a good default approach exists.

When I refer to chrome below, I mean all borders, titlebar and other window decorations.

Some initial thoughts:

- Anything which we decide should display with chrome would NOT overdraw the background of the existing window.
- Alert boxes should have chrome. They are modal and clearly are a pop-up. Without chrome they don't make sense.
- A regular window that is not a dialog (thus it is non-modal) should overdraw the entire iframe AND it should be justified to the top left of the iframe even when it is explicitly positioned on the screen.
- We need to figure out dialog boxes (which are modal). Large ones are probably best handled as normal windows (no chrome, top/left justified and overdrawing the full background). But small ones may be better with chrome and positioned relative to the owner window. For example, the date picker/calendar pane that is used in an app which expects the user to be able to see the owner window at the same time as the user interacts with the dialog. Please consider the different cases here and how we might make a good default.

What other cases can people think of here?

#2 - 12/07/2016 03:48 PM - Hynek Cihlar

Greg Shah wrote:

> What other cases can people think of here?

I am wondering whether we should somehow handle the window message area and status bar. IMHO it is a common case to have a modaless window without message area and messages output to the message area of the default window.

#3 - 12/07/2016 04:27 PM - Greg Shah

> I am wondering whether we should somehow handle the window message area and status bar. IMHO it is a common case to have a modaless window without message area and messages output to the message area of the default window.

This is a good point. And it is a tricky case. Modal dialogs grey out the background today (in embedded mode). Displaying messages in another window's message area would potentially require a common message area that always stays visible. It might look pretty weird.

#4 - 12/07/2016 04:45 PM - Hynek Cihlar

Greg Shah wrote:

> > I am wondering whether we should somehow handle the window message area and status bar. IMHO it is a common case to have a modaless window without message area and messages output to the message area of the default window.
>
> This is a good point. And it is a tricky case. Modal dialogs grey out the background today (in embedded mode). Displaying messages in another window's message area would potentially require a common message area that always stays visible. It might look pretty weird.

We could get more creative and solve this by using a proprietary control. For example the messages could be shown on demand - the user clicks a "message area" button the iframe will be filled with the current messages until the view is dismissed back to the last app window.

#5 - 01/04/2017 07:38 AM - Constantin Asofiei

Greg, the current approach is this:

1. all modal windows are left with their chrome on, regardless of their size.   An issue here is if the top-most visible window is a modal (non-alert-box) window... it should not have chrome. You mention in note 1 about large dialogs: what do we use as reference? The top-most non-modal, visible window - if it exceeds its bounds, then remove its chrome?
2. non-modal windows are re-positioned to top-left corner via javascript
3. modal windows are centered using the top-most visible non-modal window as reference
4. all non-visible windows and all windows behind the top-most non-modal window (in z-order) are hidden via CSS, using display: none (and restored the same way)

~~You mention in note 1 about large dialogs: what do we use as reference? The top-most non-modal, visible window - if it exceeds its bounds, then remove its chrome?~~

#6 - 01/04/2017 07:44 AM - Constantin Asofiei

Also, what about a nested scenario, using this z-order (from bottom to top): window1 -> dialog1 -> dialog2 -> dialog3, where dialog3 can not fit in window1 - should it be displayed without chrome, and hide window1, dialog1 and dialog2?

#7 - 01/04/2017 08:55 AM - Greg Shah

> An issue here is if the top-most visible window is a modal (non-alert-box) window... it should not have chrome.

Agreed.

> You mention in note 1 about large dialogs: what do we use as reference? The top-most non-modal, visible window -

I'm torn between using the owner window and your idea of the top-most non-modal, visible window. I can see reasons for using either one.

> if it exceeds its bounds, then remove its chrome?

I think this is close. How about we check if **either** the height or the width is equal to or greater than the reference window, then we remove its chrome. It is common for a dialog to only exceed in one dimension, but even in this case it seems likely that the user would not be expected to need to see the window below. Any such case is a chrome-removal case since it will look cleaner than doing the modal overlay thing (with chrome).

> what about a nested scenario, using this z-order (from bottom to top): window1 -> dialog1 -> dialog2 -> dialog3, where dialog3 can not fit in window1 - should it be displayed without chrome, and hide window1, dialog1 and dialog2?

Yes, I think so.

#8 - 01/04/2017 12:40 PM - Constantin Asofiei

Greg, there is a catch in deciding on a per-dialog frame case if it requires decorations or not: we can't know this until the dialog is shown; the decision can't be taken when the dialog is built/instantiated, but when is shown. And, a certain dialog can end up with or without decorations, depending when is used: are windows behind it in z-order? or is this the only window on screen?

This wouldn't be problematic if there weren't properties like BORDER-TOP-PIXELS which depend on the fact if the dialog-box has or not a title... but, wouldn't we want for these properties to report the same values, regardless if embedded mode or not?

#9 - 01/04/2017 02:46 PM - Greg Shah

> but, wouldn't we want for these properties to report the same values, regardless if embedded mode or not?

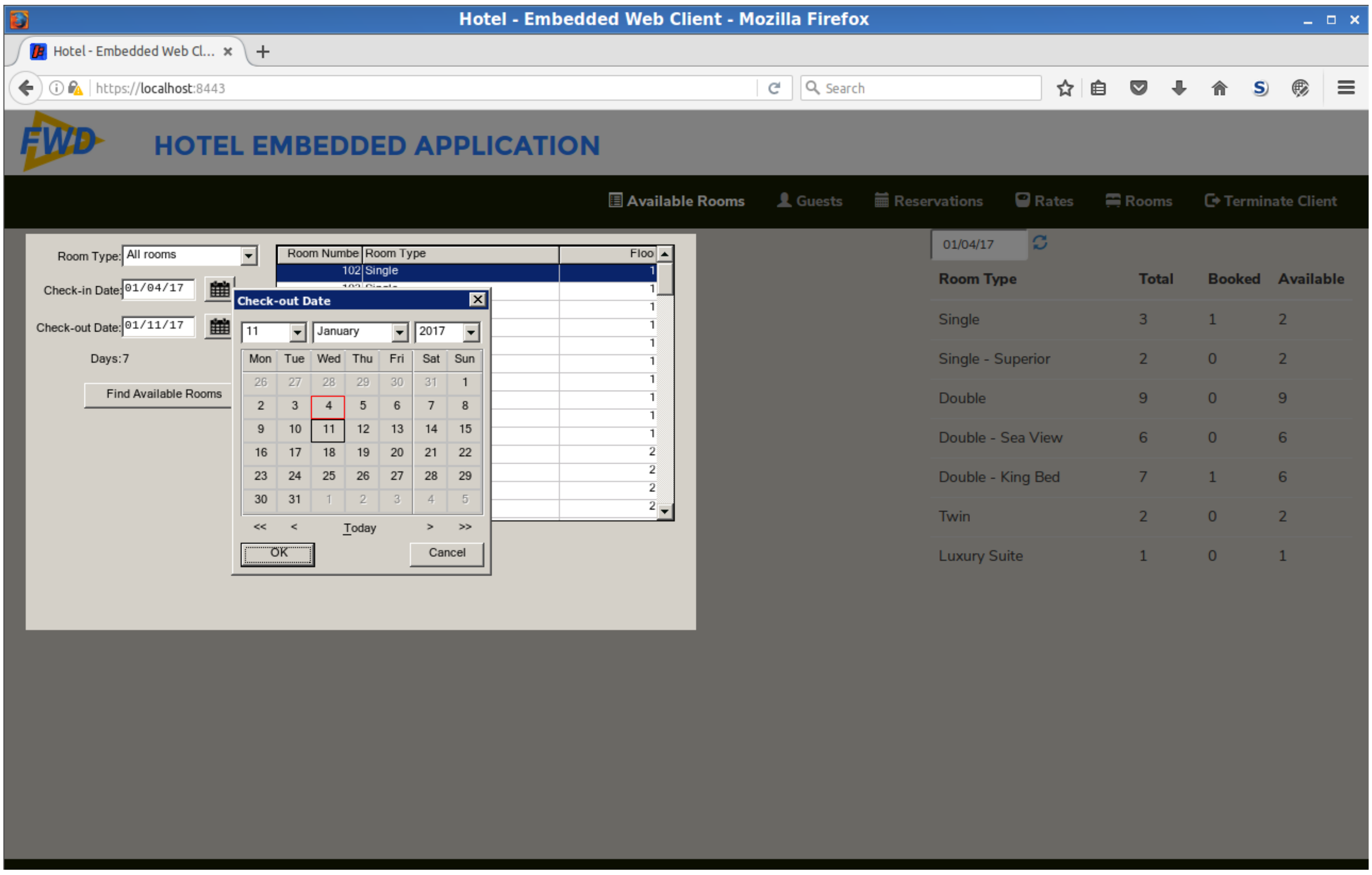
Yes, that is correct. The fact that the user can't see them doesn't matter. The 4GL code should "think" that they are there.

#10 - 01/04/2017 05:16 PM - Greg Shah

- File hotel_gui_embedded_app_modal_shading_at_owner_window.png added

Wow! I've tested out rev 11152 from 3209b. This is really close now.

The most important change I would like is to modify the overlay shading. The following example is what happens today:



Room Type: All rooms
Check-in Date: 01/04/17
Check-out Date: 01/11/17
Days:7

Room Numbe	Room Type	Floo
102	Single	1

Check-out Date

11 | January | 2017

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

01/04/17

Room Type	Total	Booked	Available
Single	3	1	2
Single - Superior	2	0	2
Double	9	0	9
Double - Sea View	6	0	6
Double - King Bed	7	1	6
Twin	2	0	2
Luxury Suite	1	0	1

Notice how the shading is around the owner (non-modal) window. It is confusing because it suggests that we should be able to interact with that window. But the idea of the shading is to be a clue to the user that this is a modal dialog. So I would like the shading to go all the way up to the modal dialog's chrome (the date picker in this case). I think this change is important for the demo.

A second request is that I would prefer to have the stacking fix so that I don't have to exclude that case from the demo. BUT if it is too much to get done then we can leave it out.

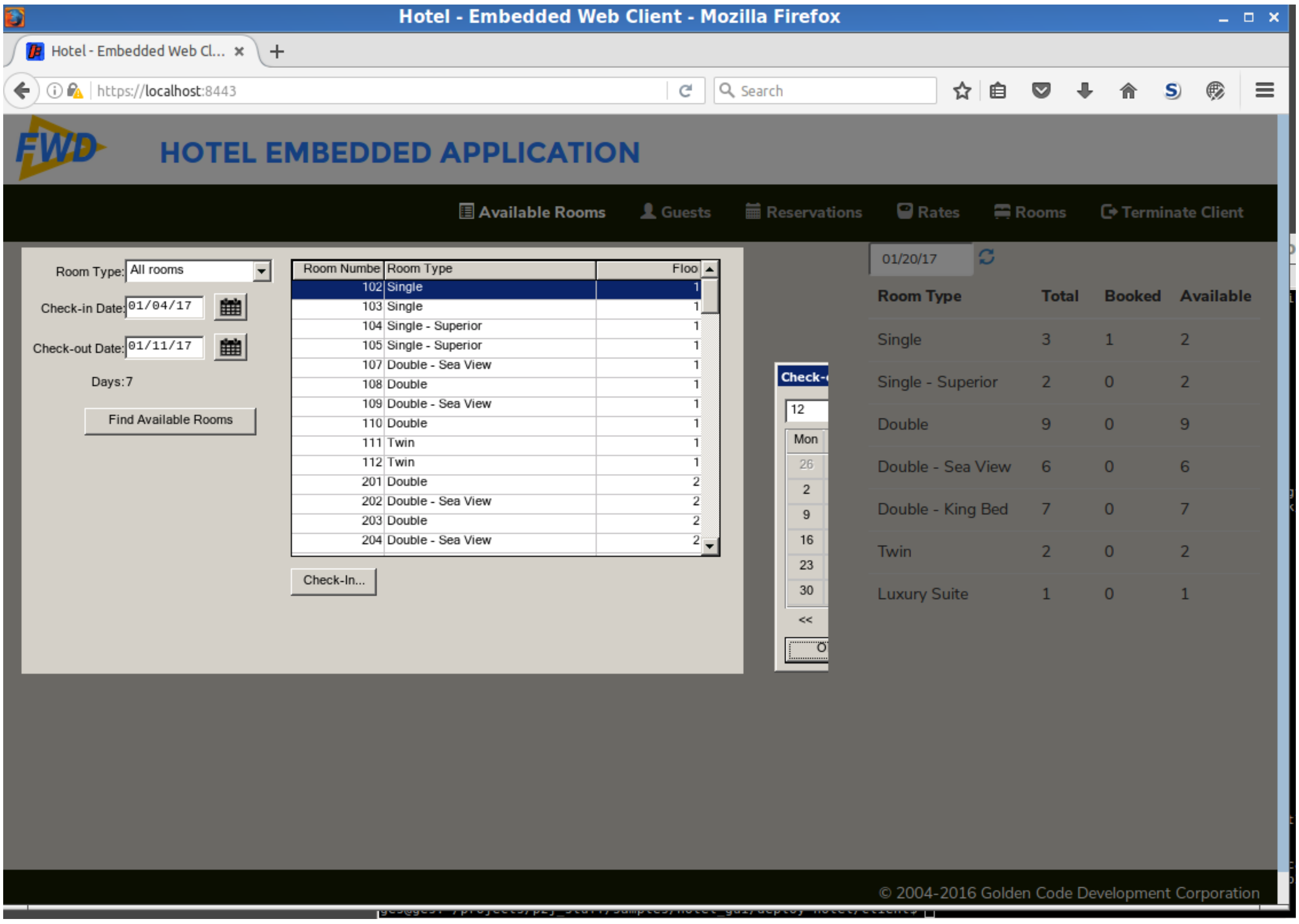
#11 - 01/04/2017 05:20 PM - Greg Shah

- File hotel_gui_embedded_modal_dialog_disappears_on_bottom.png added

- File hotel_gui_embedded_modal_dialog_disappears_on_side.png added

Item for discussion (BUT NOT needed for the demo): when you move a modal dialog, it can go "underneath" the outside edges of the iframe, which seems to be not useful. It also would probably be confusing for a user.

Examples:



Room Type: All rooms

Check-in Date: 01/04/17

Check-out Date: 01/11/17

Days:7

Room Numbe	Room Type	Floo
102	Single	1
103	Single	1
104	Single - Superior	1
105	Single - Superior	1
107	Double - Sea View	1
108	Double	1
109	Double - Sea View	1
110	Double	1
111	Twin	1
112	Twin	1
201	Double	2
202	Double - Sea View	2
203	Double	2
204	Double - Sea View	2

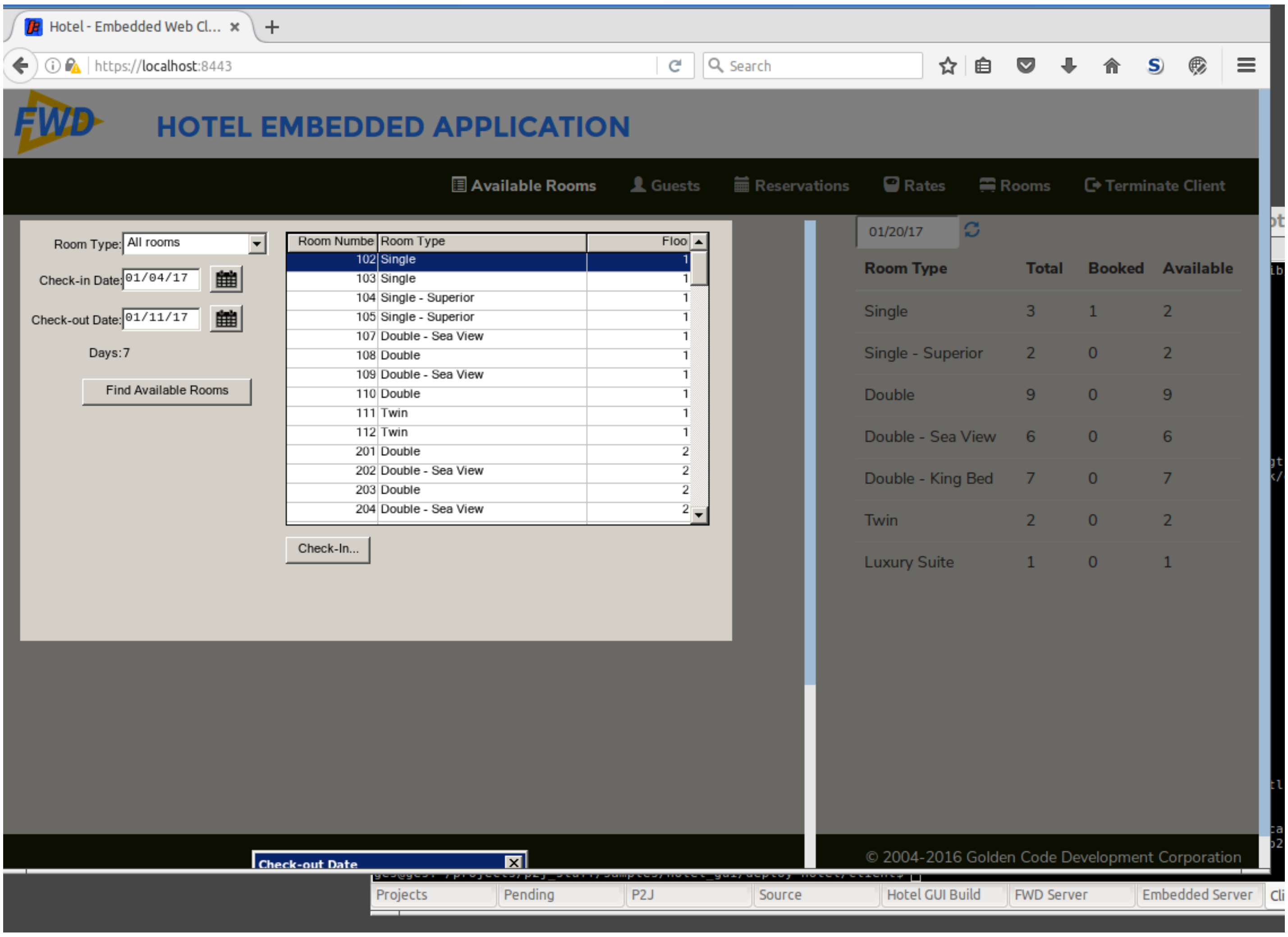
12

Mon
26
2
9
16
23
30

01/20/17

Room Type	Total	Booked	Available
Single	3	1	2
Single - Superior	2	0	2
Double	9	0	9
Double - Sea View	6	0	6
Double - King Bed	7	0	7
Twin	2	0	2
Luxury Suite	1	0	1

© 2004-2016 Golden Code Development Corporation



Room Type: All rooms

Check-in Date: 01/04/17

Check-out Date: 01/11/17

Days:7

Room Numbe	Room Type	Floo
102	Single	1
103	Single	1
104	Single - Superior	1
105	Single - Superior	1
107	Double - Sea View	1
108	Double	1
109	Double - Sea View	1
110	Double	1
111	Twin	1
112	Twin	1
201	Double	2
202	Double - Sea View	2
203	Double	2
204	Double - Sea View	2

01/20/17

Room Type	Total	Booked	Available
Single	3	1	2
Single - Superior	2	0	2
Double	9	0	9
Double - Sea View	6	0	6
Double - King Bed	7	0	7
Twin	2	0	2
Luxury Suite	1	0	1

© 2004-2016 Golden Code Development Corporation

Eric suggests disabling the ability of the user to grab and move the window using the titlebar. I guess in this case it would just never allow dragging.

As mentioned above, it is not needed in time for the demo. I just want to include it in the discussion.

#12 - 01/05/2017 05:34 AM - Hynek Cihlar

Greg Shah wrote:

> Eric suggests disabling the ability of the user to grab and move the window using the titlebar. I guess in this case it would just never allow dragging.

I think this would degrade usability, there are valid cases when you want to move the window away, for example to see the information in its owner window.

We can limit the position so that the window never leaves its screen area. This behavior would have to be conditional, in Swing for example it is valid to have any window leave the screen.

#13 - 01/05/2017 05:44 AM - Constantin Asofiei

Greg Shah wrote:

> Notice how the shading is around the owner (non-modal) window. It is confusing because it suggests that we should be able to interact with that window. But the idea of the shading is to be a clue to the user that this is a modal dialog. So I would like the shading to go all the way up to the modal dialog's chrome (the date picker in this case). I think this change is important for the demo.

I've fixed this in 3209b rev 11153. I've added an internal overlay panel (in the P2J window...), to cover the P2J windows behind the topmost modal one. Note that there is an issue here: as there are two overlay panels, one internal in the iframe and one external in the embedded window, where they overlap (in the iframe) the color is darker; I don't know if this can be solved.

> A second request is that I would prefer to have the stacking fix so that I don't have to exclude that case from the demo. BUT if it is too much to get done then we can leave it out.

I have an idea here: let every window draw with the chrome, and fix it in the javascript - get the boundary of the internal content (without chrome) and force the canvas to have the size and content as if no chrome exists. This removes the decoration logic from the P2J runtime and moves it at the driver. If this works, I think will be cleaner.

#14 - 01/05/2017 07:46 AM - Greg Shah

Hynek Cihlar wrote:

> Greg Shah wrote:
>
>> Eric suggests disabling the ability of the user to grab and move the window using the titlebar. I guess in this case it would just never allow dragging.
>
> I think this would degrade usability, there are valid cases when you want to move the window away, for example to see the information in its owner window.
>
> We can limit the position so that the window never leaves its screen area. This behavior would have to be conditional, in Swing for example it is valid to have any window leave the screen.

Yes, this makes sense.

#15 - 01/05/2017 08:17 AM - Greg Shah

> I've fixed this in 3209b rev 11153. I've added an internal overlay panel (in the P2J window...), to cover the P2J windows behind the topmost modal one. Note that there is an issue here:

This only works the first time a modal dialog is brought up. After that it never shows again.

Also, at least one time (but not other times) I have seen the other screens (e.g. Guests) fail to load after using the date picker a few times on the first screen (Available Rooms).

> as there are two overlay panels, one internal in the iframe and one external in the embedded window, where they overlap (in the iframe) the color is darker; I don't know if this can be solved.

Wasn't that darker color also showing in the original approach? I think it is just being drawn darker anyway, this is the area of the iframe that is outside of the main window and is supposed to be drawn with the same background color as the 4GL window background. It is important to try to fix this as it is visually distracting.

#16 - 01/05/2017 08:20 AM - Greg Shah

Interestingly, when you nest dialogs, the inner shading "comes back" (after being missing).

From Guests, use the Update button to get to the Update Reservations dialog, then use the Update button on that window.

Perhaps the missing inner shading is due to the stacking issue.

#17 - 01/05/2017 08:23 AM - Greg Shah

> I have an idea here: let every window draw with the chrome, and fix it in the javascript - get the boundary of the internal content (without chrome) and force the canvas to have the size and content as if no chrome exists. This removes the decoration logic from the P2J runtime and moves it at the driver. If this works, I think will be cleaner.

I don't have a major objection to this. But I do think it is possible that users may want to embed the Swing client in a larger application. One recent customer thought about doing this as a fall back plan (where they would have the P2J client embedded in and integrated with their Java application).

Having the logic in common code would be helpful for that case. I do realize the feature is not exposed to the Swing client today.

#18 - 01/05/2017 08:49 AM - Constantin Asofiei

Greg Shah wrote:

> Wasn't that darker color also showing in the original approach? I think it is just being drawn darker anyway, this is the area of the iframe that is outside of the main window and is supposed to be drawn with the same background color as the 4GL window background. It is important to try to fix this as it is visually distracting.

No, is not that, because I've disabled it

> Interestingly, when you nest dialogs, the inner shading "comes back" (after being missing).
> Perhaps the missing inner shading is due to the stacking issue.

The problem is with computing the z-index in removeZOrderEntryWorker - AFAIK all windows need to have different z-order values, but this allows two windows to have the same z-order. I'm trying to fix it.

> I don't have a major objection to this. But I do think it is possible that users may want to embed the Swing client in a larger application. One recent customer thought about doing this as a fall back plan (where they would have the P2J client embedded in and integrated with their Java application).
> Having the logic in common code would be helpful for that case. I do realize the feature is not exposed to the Swing client today.

OK, I'll try to think to a more generic approach... the idea would be the same, the driver will display on the undecorated content.

#19 - 01/05/2017 09:36 AM - Constantin Asofiei

Greg, see 3209b rev 11154 - it fixes the overlay panel problems (not showing sometimes and the bgcolor).

#20 - 01/05/2017 09:57 AM - Greg Shah

It works perfectly! I really, really like it.

What is left on this task?

#21 - 01/05/2017 10:29 AM - Constantin Asofiei

Greg Shah wrote:

> What is left on this task?

1. how to handle status-area and message-area - haven't thought of a good solution for this yet - Hynek's idea is interesting, but the impl details are ... tricky.
2. generic approach to solve the dialog-box without chrome issue (when is top-level, when is wider/taller than the parent window, etc)
3. do not allow the dialog-boxes with chrome to be moved outside the iframe

#22 - 01/05/2017 11:48 AM - Greg Shah

> how to handle status-area and message-area - haven't thought of a good solution for this yet - Hynek's idea is interesting, but the impl details are ... tricky.

You're referring to his idea in note 4? I like this idea, but would prefer not to put the control in the user's hands. The embedded app itself can control visibility of the proprietary control that displays these messages. Perhaps that would take away some of the trickiness.

If not using a proprietary control, then an alternative would be to provide them to the embedded app and let each app decide how/when to display the messages. The embedded app could register a callback that we can call every time new content would be displayed there. We might want to provide other info (whether it was a status or message, what window...), but that could easily be handled with parameters to the callback.

#23 - 01/05/2017 11:49 AM - Greg Shah

The callback approach could be done quickly and then we could defer the proprietary control as an improvement for later.

#24 - 01/05/2017 12:36 PM - Constantin Asofiei

Greg Shah wrote:

> The callback approach could be done quickly and then we could defer the proprietary control as an improvement for later.

I agree, the callback looks better.

#25 - 01/05/2017 01:40 PM - Greg Shah

How much time do you need to finish this task?

#26 - 01/05/2017 02:50 PM - Constantin Asofiei

Greg Shah wrote:

> How much time do you need to finish this task?

I think tomorrow will be finished.

#27 - 01/05/2017 04:46 PM - Constantin Asofiei

Does anyone know how to get the iframe height (and not the screen height...) using javascript? I've tried document.body.scrollHeight executed from the iframe side, but this doesn't work in firefox - it returns 0... in chrome it works.

This is needed to not allow the dialog to be moved vertically beyond the iframe's bottom.

#28 - 01/06/2017 05:53 PM - Constantin Asofiei

Something I haven't considered when translating the top-left origin and resizing the canvas, so that the chrome gets 'hidden': all mouse processing data, combo drop-down windows, right-click menus, etc, will use the untranslated/original coordinates... this complicates things a lot, as these would need to be intercepted and coordinates be adjusted. I'll try to think of a smart/easy way to intercept and adjust these coordinates.

To keep both legacy attributes (like border-related) untouched, one idea would be to color the titlebar/window border/etc with the window's bgcolor, so they are not actually visible, but they will still occupy space. Keeping these from responding on events is another tricky area.

#29 - 01/06/2017 06:32 PM - Greg Shah

Hmm. Good points. Is it worth deferring these pieces while we have more urgent work to complete?

#30 - 01/06/2017 06:42 PM - Constantin Asofiei

Greg Shah wrote:

> Hmm. Good points. Is it worth deferring these pieces while we have more urgent work to complete?

Beside adjusting the events/etc to consider that the chrome is not there, what is left is to 'hook up' the decoration verification code in these methods at the driver and testing:

```
resizeWindow(int, int)
setWindowBounds(int, int, int, int)
setWindowLocation(int, int)
```

```
setWindowVisible(boolean)
setWindowEnabled(boolean)
stackWindows(int[])
moveToTop(int, boolean)
moveToBottom(int)
```

Currently the code is added only for GuiWebDriver.setWindowVisible.

I've created 3218a for these changes, you can take a look if you want.

I'll check on Monday if I can find a quick way to solve this; otherwise, I can switch to other work.

#31 - 03/07/2017 12:39 PM - Greg Shah

- Related to Feature #3258: improve javascript upcalls from the embedded client added

#32 - 03/07/2017 12:45 PM - Greg Shah

- Related to Feature #3259: add useful web components that can be easily integrated into the embedded web client added

#33 - 03/07/2017 01:19 PM - Greg Shah

- Related to Feature #3260: auto-conversion and enablement for the embedded web client added

#34 - 03/07/2017 01:24 PM - Greg Shah

- Related to Feature #3261: enhanced browse that can optionally selected as a replacement for the default ABL browse added

#35 - 02/07/2018 11:25 AM - Greg Shah

What is the status of this task? Do we have anything to do or have the issues been resolved by changes elsewhere?

#36 - 02/07/2018 05:15 PM - Constantin Asofiei

Greg Shah wrote:

> What is the status of this task? Do we have anything to do or have the issues been resolved by changes elsewhere?

3218a was WIP and not finalized. I need to re-review it to recall the state of the changes and what is missing, but as I recall there were some issues with opening another window from the 'main' embedded window (how deep to stack it, do we allow breadcrumbs to go back in the stack, some issues with removing decorations, what happens with large alert boxes, etc).

#37 - 08/10/2018 01:24 PM - Constantin Asofiei

From 3218a branch, all usable code was extracted into #3469 branches and used. Otherwise, the branch can be archived.

#38 - 08/10/2018 04:31 PM - Greg Shah

I think this task can be closed, right?

#39 - 08/10/2018 05:32 PM - Constantin Asofiei

Greg Shah wrote:

> I think this task can be closed, right?

Yes. 3218a was archived as dead.

#40 - 08/12/2018 09:51 PM - Greg Shah

- % Done changed from 0 to 100

- Status changed from WIP to Closed

Files

hotel_gui_embedded_app_modal_shading_at_owner_window.png	80.6 KB	01/04/2017	Greg Shah
hotel_gui_embedded_modal_dialog_disappears_on_side.png	89.9 KB	01/04/2017	Greg Shah
hotel_gui_embedded_modal_dialog_disappears_on_bottom.png	85.1 KB	01/04/2017	Greg Shah