

User Interface - Bug #3548

Window widget FULL-(WIDTH|HEIGHT)-(CHARS|PIXELS) proper initialization

04/12/2018 04:33 PM - Eugenie Lyzenko

Status: Closed	Start date:
Priority: Normal	Due date:
Assignee: Eugenie Lyzenko	% Done: 100%
Category:	Estimated time: 0.00 hour
Target version:	case_num:
billable: No	
vendor_id: GCD	
Description	
Related issues:	
Related to User Interface - Bug #3550: Window restore from maximized state GU...	New 04/18/2018

History

#1 - 04/12/2018 04:34 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

It seems the lifetime of the FULL-WIDTH-CHARS value is a bit more complex. I did a few simple tests and here are some early findings:

- FULL-* attribute returns unknown value for a new window (after create window h) before it is made visible.
- FULL-* attribute returns valid value for a new window after it is made visible.
- FULL-* attribute returns a valid value default-window, even before it is made visible. I tested this with running the procedure file directly with prowin without showing the procedure editor.

So I'm not sure the init() method is the right place.

#3 - 04/12/2018 05:52 PM - Greg Shah

- Status changed from New to WIP

- Assignee set to Eugenie Lyzenko

- Start date deleted (04/12/2018)

#4 - 04/12/2018 06:15 PM - Eugenie Lyzenko

Created task branch 3548a from trunk revision 11246.

#5 - 04/12/2018 07:35 PM - Eugenie Lyzenko

Currently I see two deviation for FWD comparing to 4GL:

1. The default window attribute values are unknown (?) when should be properly initialized. No matter if it is visible or not.
2. The attribute values are valid for created but not yet visible dynamic window when the correct value is unknown (?).

The testcase is win_full_hw_attr.p.

#6 - 04/13/2018 12:57 PM - Eugenie Lyzenko

Task branch 3548a has been rebased with recent trunk and updated for review to revision 11248.

This is the base fix for deviations found for this issue. The default window set up early enough (even before it is visible). The dynamic windows has now unknown values before realization and valid values when become first time visible.

Now testing in Windows OS to get correct values to FULL-* attributes. To exactly compare what values we have in both 4GL and FWD.

#7 - 04/13/2018 09:43 PM - Eugenie Lyzenko

Task branch 3548a has been updated for review to revision 11249.

Two more changes here:

1. I've accidentally found the abend issue (-2 as array index exception) in EditorGuiImpl for mouse click.

2. The issue related change is to eliminate deviations for full size window attributes. New method has been added to separate calculation. It works good for dynamic window that has all internal defined before update full size attributes (just before first painting). However the default window behaves differently and valid FULL-* attributes become problematic at the moment when window is not yet visible. We need to set up the values when not available all decorations have been initialized. The status line and message area are the examples. The FULL-* attributes are computing in ThinClient.initializePost() code and we can either take into account some average values for status and message heights or to choose another moment to compute FULL-* attributes for default window. Continue investigation.

#8 - 04/16/2018 07:02 AM - Eugenie Lyzenko

Task branch 3548a has been updated for review to revision 11250.

This is the small typo fixes for file headers info records.

#9 - 04/16/2018 01:30 PM - Hynek Cihlar

Code review task branch 3548a revision 11250.

WindowGuiImpl.fullSizeAttributesUpdated is redundant. Just move the call updateFullSizeAttributes to WindowGuiImpl.init at the end of the body and protect it with if (!getId().equals(WidgetId.DEFAULT_WINDOW_ID)).

WindowChuiImpl.initDefaults() is redundant, there is already no-op body Window.initDefaults().

ThinClient.initializePost(), default window may be used before the environment is fully initialized (see DefaultLoginPanel for example) so the call to initDefaults() should be better moved to WindowManager.getDefaultWindow().

#10 - 04/16/2018 01:33 PM - Eugenie Lyzenko

Task branch 3548a has been updated for review to revision 11251.

This is reworked code with changed location to get default window FULL-* attributes. No changes in ChUI code, no changes in ThinClient - all this

was undone. Only GUI code is now affected. And this is release candidate on my guess.

#11 - 04/17/2018 04:48 AM - Hynek Cihlar

Code review 3548a revision 11251.

The changes are good. But note that the value of the FULL-* attribute changes when the window layout changes. For example when a menubar is made visible or hidden.

#12 - 04/17/2018 07:10 AM - Eugenie Lyzenko

Hynek Cihlar wrote:

Code review 3548a revision 11251.

The changes are good. But note that the value of the FULL-* attribute changes when the window layout changes. For example when a menubar is made visible or hidden.

Task branch 3548a has been updated for review to revision 11252.

The idea is to refresh full size attributes for the changes you mentioned. I guess FWD should force doLayout() in this case for window GUI widget. Applied to static or visible windows only.

#13 - 04/17/2018 04:56 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

Hynek Cihlar wrote:

Code review 3548a revision 11251.

The changes are good. But note that the value of the FULL-* attribute changes when the window layout changes. For example when a menubar is made visible or hidden.

Task branch 3548a has been updated for review to revision 11252.

The idea is to refresh full size attributes for the changes you mentioned. I guess FWD should force doLayout() in this case for window GUI widget. Applied to static or visible windows only.

Code review 3548a revision 11252.

The code changes are OK except the !config().dynamic in WindowGuiImpl.doLayout(). This makes the behavior of the FULL- attributes different for static (default) and dynamic windows. Consider the case when the window layout is changed (for example menu-bar assigned) while the window is not visible.

#14 - 04/17/2018 05:13 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

Code review 3548a revision 11252.

The code changes are OK except the `!config().dynamic` in `WindowGuiImpl.doLayout()`. This makes the behavior of the FULL- attributes different for static (default) and dynamic windows.

Static and dynamic windows are really handling differently. Dynamic windows do not have FULL-* attributes defined until becoming visible.

Consider the case when the window layout is changed (for example menu-bar assigned) while the window is not visible.

In this case static window will update FULL-* attributes, while dynamic - not.

What is wrong here?

#15 - 04/17/2018 05:19 PM - Hynek Cihlar

Eugenie Lyzenko wrote:

Consider the case when the window layout is changed (for example menu-bar assigned) while the window is not visible.

In this case static window will update FULL-* attributes, while dynamic - not.

What is wrong here?

The test case is as follows. Show a window, hide it, assign menu-bar, read the FULL- attributes (this is where the behavior differs in 3548a; in the native app FULL- attributes report the same values for default-window and dynamic windows), show the window.

#16 - 04/17/2018 05:36 PM - Eugenie Lyzenko

Hynek Cihlar wrote:

The test case is as follows. Show a window, hide it, assign menu-bar, read the FULL- attributes (this is where the behavior differs in 3548a; in the native app FULL- attributes report the same values for default-window and dynamic windows), show the window.

Task branch 3548a has been updated for review to revision 11253.

OK. Must be fixed with this update.

#17 - 04/18/2018 02:40 AM - Hynek Cihlar

Code review 3548a revision 11253.

The changes are OK. I don't register any more open points for the branch.

#18 - 04/18/2018 07:09 AM - Greg Shah

What GUI testing is needed to confirm this is safe for trunk? Is it enough to test the AW and Hotel usage cases?

#19 - 04/18/2018 07:25 AM - Eugenie Lyzenko

Greg Shah wrote:

What GUI testing is needed to confirm this is safe for trunk? Is it enough to test the AW and Hotel usage cases?

I think it is enough. The standalone test and AW is working OK.

#20 - 04/18/2018 07:26 AM - Greg Shah

Have you already tested Hotel GUI?

#21 - 04/18/2018 07:35 AM - Eugenie Lyzenko

Greg Shah wrote:

Have you already tested Hotel GUI?

In progress now. Need to check one option comparing with trunk. Up to 30 min. more I guess.

#22 - 04/18/2018 08:01 AM - Eugenie Lyzenko

Not found the regressions in hotel GUI.

But found issue in trunk for hotel GUI. When window is restoring from maximized state by restore title bar button the window size is not changing but location is changed to become negative so the top-left corner of the window is moving outside the visible area.

#23 - 04/18/2018 08:25 AM - Greg Shah

Please merge 3548a to trunk.

But found issue in trunk for hotel GUI. When window is restoring from maximized state by restore title bar button the window size is not changing but location is changed to become negative so the top-left corner of the window is moving outside the visible area.

Please create a new task for this and link it here.

#24 - 04/18/2018 09:32 AM - Eugenie Lyzenko

Branch 3548a was merged to trunk as revno 11248 then it was archived.

#25 - 04/18/2018 09:34 AM - Eugenie Lyzenko

- *Related to Bug #3550: Window restore from maximized state GUI mode issue added*

#26 - 04/18/2018 09:56 AM - Greg Shah

Did you test Hotel GUI in both Swing and Web?

#27 - 04/18/2018 10:22 AM - Eugenie Lyzenko

Greg Shah wrote:

Did you test Hotel GUI in both Swing and Web?

Yes. The Swing difference is in new issue that found. Swing window (size and location) is not changing comparing to maximized state.

#28 - 04/18/2018 10:29 AM - Greg Shah

- *Status changed from WIP to Closed*

- *% Done changed from 0 to 100*