User Interface - Feature #3284

implement more window widget support

04/27/2017 02:29 PM - Greg Shah

Status:	Closed	**Start date:**	04/27/2017
Priority:	Normal	**Due date:**	
Assignee:	Hynek Cihlar	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD
Description			

History

#1 - 04/27/2017 02:29 PM - Greg Shah

Per #3257-9, add the following window widget attributes:

WINDOW-STATE (finish the runtime)
MIN-BUTTON
MAX-BUTTON
SHOW-IN-TASKBAR
CONTROL-BOX
ALWAYS-ON-TOP
SMALL-TITLE

#2 - 06/20/2017 07:57 AM - Hynek Cihlar

The WINDOW-DELAYED-MINIMIZE window state runtime behavior is **really** bizarre. The documentation says "The window is minimized (iconified) the next time a new window, dialog box, or alert box is displayed. This differs from setting WINDOW-STATE to WINDOW-MINIMIZED, which minimizes the window immediately.", but the actual behavior is a lot more fun!

When the delayed minimize state is set **before** the window has been realized, the WINDOW-STATE **does** report WINDOW-DELAYED-MINIMIZE, even when it gets realized, however the window behaves as if WINDOW-STATE was set to WINDOW-NORMAL (no delayed minimize whatsoever).

When the delayed minimize state is set **after** the window has been realized, the WINDOW-STATE **does not** report WINDOW-DELAYED-MINIMIZE, it reports the value previously set in WINDOW-STATE. However it does exhibit a minimize behavior such that it minimizes when another window (or dialog, or alert-box) gets first shown **and** gets first focused **and** the new window was shown **after** the WINDOW-DELAYED-MINIMIZE was set.

#3 - 06/20/2017 08:48 AM - Greg Shah

> but the actual behavior is a lot more fun!

:)

Good work isolating the behavior.

#4 - 06/20/2017 04:32 PM - Hynek Cihlar

Created task branch 3284a and committed runtime support for WINDOW-STATE:WINDOW-DELAYED-MINIMIZE to it.

#5 - 08/02/2017 04:55 PM - Hynek Cihlar

3284a rebased against trunk 11156.

#6 - 08/09/2017 06:38 AM - Hynek Cihlar

Regarding the SHOW-IN-TASKBAR support, I didn't find a reliable and portable way to make a top-level window not appear in the taskbar. There are two possible ways, but none of them works 100%.

(1) Use java.awt.Window#setType(java.awt.Frame.Type.UTILITY). This however only works in Windows, but not on Linux (I am not sure about MacOS).

(2) Use JDialog instead of JFrame. This is not ideal solution either. First the task-bar icon is hidden only when the JDialog's parent window is itself visible (so fails for cases when only single 4GL top-level window is visible) and second JDialog windows introduce additional window-hierarchy behavior which contradicts the 4GL semantics.

Due to the above and since the Swing support AFAIK is not that critical I suggest to leave the window icon in the task-bar for Swing GUI driver and implement the rest of the SHOW-IN-TASKBAR behavior as is.

#7 - 08/10/2017 08:53 AM - Greg Shah

> Due to the above and since the Swing support AFAIK is not that critical I suggest to leave the window icon in the task-bar for Swing GUI driver and implement the rest of the SHOW-IN-TASKBAR behavior as is.

I agree. Do the following:

1. Make sure it is clearly documented in the code.
2. Add a task, link it here and explain the limitation.
3. When you update the gap analysis rules, set the runtime value to partial and put a comment on the line for this attribute.

#8 - 08/16/2017 03:14 PM - Hynek Cihlar

Rebased 3284a against current trunk.

#9 - 08/17/2017 07:01 PM - Hynek Cihlar

Rebased 3284a against trunk 11158.

#10 - 08/22/2017 03:40 PM - Greg Shah

Code Review Task Branch 3284a Revision 11189

1. In methods_attributes.rules, I don't think you need to use and htype == prog.kw_session for the STARTUP-PARAMETERS and SUPPRESS-WARNINGS attributes. They can only be used with the SESSION handle, so the qualifier is not needed. It is best to remove it because detecting that the usage is the SESSION handle cannot be done 100%. For example, if the SESSION is assigned to a handle and then passed as a parameter to a procedure, we can't tell if it is the SESSION handle.

2. In methods_attributes.rules, the window additions should be made in load_descriptors and they should be in alpha order.

3. Why is GET-MOUSE-POINTER defined as METH_POLY in SignatureHelper?

4. Why is GET-MOUSE-POINTER defined as a keyword KW_GET_MCP? I don't see where the "C" comes from.

5. progress.g should only have a single history entry.

6. Add // FWD extension, not real 4GL! as a comment for the KW_GET_MCP keyword def and attribute defin progress.g (linee 3501 and 7107).

7. I think the javadoc return definition for getMousePosition is incorrect. It states: mouse cursor position, x coordinate at index 0, y coordinate at index 0 but surely either x or y is at index 1? See ClientExports.

8. In CommonWindow, some setter methods are missing. There should be both boolean and logical forms for SHOW-IN-TASKBAR, MIN-BUTTON and MAX-BUTTON.

9. Assuming there is common behavior for unknown value handling, then default methods could be used to implement all the new setters that take logical. Then the widget implementation classes will only need to provide the boolean form. If unknown is handled differently, then the boolean form can be default in CommonWindow. This is a good practice even if there is only 1 implementer of the interface, since if there are ever implementers added later it is less to implement.

10. Why is com.goldencode.p2j.util.Text; being imported in GenericWidget?

11. Window, WindowManager, CaptionButton, CaptionButtonType, GuiWindow, WindowLayout, WindowTitleBar, WindowTitlePopupGuiImpl, AbstractGuiDriver, GuiOutputDriver, GuiWebEmulatedWindow, ClassicTheme, Windows10Theme, Windows8Theme are missing history entries.

12. What is the origin of the Window.pushConfig() changes? They seem reasonable, but they are also in a pretty sensitive area and it is not 100% clear that they are safe.

13. WindowManager.registerDelayedMinimize(), WindowManager.unregisterDelayedMinimize(), WindowLayout.getSlotsPerRow(), WindowTitleBar.allButtons are missing javadoc.

14. I think the changes in AbstractClientDialog and OverlayWindow can be reverted.

#11 - 08/22/2017 03:56 PM - Hynek Cihlar

Greg Shah wrote:

> Code Review Task Branch 3284a Revision 11189
>
> 1. In methods_attributes.rules, I don't think you need to use and htype == prog.kw_session for the STARTUP-PARAMETERS and SUPPRESS-WARNINGS attributes. They can only be used with the SESSION handle, so the qualifier is not needed. It is best to remove it because detecting that the usage is the SESSION handle cannot be done 100%. For example, if the SESSION is assigned to a handle and then passed as a parameter to a procedure, we can't tell if it is the SESSION handle.

Will fix.

> 2. In methods_attributes.rules, the window additions should be made in load_descriptors and they should be in alpha order.

Will fix.

> 3. Why is GET-MOUSE-POINTER defined as METH_POLY in SignatureHelper?

What type to use? The method returns BaseDataType[].

> 4. Why is GET-MOUSE-POINTER defined as a keyword KW_GET_MCP? I don't see where the "C" comes from.

I originally named the method GET-MOUSE-CURSOR-POINTER. But I shortened it and since didn't update the constant. Will fix.

> 5. progress.g should only have a single history entry.

Will fix.

> 6. Add // FWD extension, not real 4GL! as a comment for the KW_GET_MCP keyword def and attribute defin progress.g (linee 3501 and 7107).

Will fix.

> 7. I think the javadoc return definition for getMousePosition is incorrect. It states: mouse cursor position, x coordinate at index 0, y coordinate at index 0 but surely either x or y is at index 1? See ClientExports.

Will fix.

> 8. In CommonWindow, some setter methods are missing. There should be both boolean and logical forms for SHOW-IN-TASKBAR, MIN-BUTTON and MAX-BUTTON.

Will add.

> 9. Assuming there is common behavior for unknown value handling, then default methods could be used to implement all the new setters that take logical. Then the widget implementation classes will only need to provide the boolean form. If unknown is handled differently, then the boolean form can be default in CommonWindow. This is a good practice even if there is only 1 implementer of the interface, since if there are ever implementers added later it is less to implement.

Will fix.

> 10. Why is com.goldencode.p2j.util.Text; being imported in GenericWidget?

This is the IDE to blame, Text class is referenced in the file, but the import is redundant. Will remove.

> 11. Window, WindowManager, CaptionButton, CaptionButtonType, GuiWindow, WindowLayout, WindowTitleBar, WindowTitlePopupGuiImpl, AbstractGuiDriver, GuiOutputDriver, GuiWebEmulatedWindow, ClassicTheme, Windows10Theme, Windows8Theme are missing history entries.

These are yet unfinished changes for 3284.

> 12. What is the origin of the Window.pushConfig() changes? They seem reasonable, but they are also in a pretty sensitive area and it is not 100% clear that they are safe.

The changes there fix an NPE and the case when ASSIGN is used in CREATE WINDOW statement. Without the change, the config would not reflect the @ASSIGN@ed values.

> 13. WindowManager.registerDelayedMinimize(), WindowManager.unregisterDelayedMinimize(), WindowLayout.getSlotsPerRow(), WindowTitleBar.allButtons are missing javadoc.

These are yet unfinished changes for 3284.

> 14. I think the changes in AbstractClientDialog and OverlayWindow can be reverted.

Will fix.

#12 - 08/22/2017 04:04 PM - Greg Shah

> > 3. Why is GET-MOUSE-POINTER defined as METH_POLY in SignatureHelper?
>
> What type to use? The method returns BaseDataType[].

I guess you can leave it for now. In the method implementation, why not return integer[] instead of BDT[]?

#13 - 08/22/2017 04:14 PM - Hynek Cihlar

Greg Shah wrote:

> In the method implementation, why not return integer[] instead of BDT[]?

No good reason I could come up with, I will change it to integer[].

#14 - 08/23/2017 09:29 PM - Hynek Cihlar

3284a revision 11175 resolves most of this issue except:

- runtime support of ALWAYS-ON-TOP
- when window is minimized it should not handle legacy input events
- Windows8 and Windows10 small close window caption icons are not pixel perfect, in fact, these are best-effort guesses how Progress renders the icons on these systems.

Greg, if you agree, I will start regression testing of 3284a without the open items above and deal with them in a new branch.

The revision also resolves the feedback from review in note 10.

Please review.

#15 - 08/24/2017 08:29 AM - Greg Shah

Code Review Task Branch 3284a Revision 11175

I like the use of default in the window hierarchy. The window state implementation is much cleaner too, which is also good.

1. I did have to restore the session handle checking to the kw_3d processing in methods_attributes.rules. That one was already there and is needed because kw_3d can be used on other handle types besides session.

2. Please rebase.

3. WindowManager.LOG, WindowGuiImpl.windowState are missing javadoc.

#16 - 08/24/2017 08:34 AM - Greg Shah

> I will start regression testing of 3284a without the open items above and deal with them in a new branch.

Yes, it is really urgent to get this update into the trunk for the conversion pieces. For the runtime changes, it is important to ensure no regressions. But it is OK to do runtime work in a new branch.

Please address the (minor) items in #3284-15 and then go into testing. Besides ChUI conversion and runtime testing, please do careful GUI testing of Swing and web client for Hotel GUI, the original customer GUI application and any standalone GUI tests that are window related.

#17 - 08/24/2017 10:19 AM - Hynek Cihlar

3284a rebased against trunk revision 11160.

#18 - 08/25/2017 04:42 PM - Hynek Cihlar

Removed all cumulative runtime changes from 3284a and kept only conversion changes. The removed changes will go to 3284b. Rebased 3284a against trunk 11162.

#19 - 08/25/2017 04:48 PM - Greg Shah

I think there is a problem with the branch. It has quite a bit of mess from the rebase.

#20 - 08/25/2017 04:57 PM - Hynek Cihlar

Greg Shah wrote:

> I think there is a problem with the branch. It has quite a bit of mess from the rebase.

Sorry I post the rebase notification too early yet during the bzr push. It's pushed now.

#21 - 08/25/2017 05:11 PM - Greg Shah

Code Review Task branch 3284a Revision 11185

I'm fine with the changes.

One question: Shouldn't the gap marking for window-state runtime remain "partial"? It was previously completely implemented except for delayed minimize.

Get this conversion regression tested and it can be merged to trunk (with the gap analysis fix included).

#22 - 08/25/2017 05:19 PM - Hynek Cihlar

Hynek Cihlar wrote:

> Removed all cumulative runtime changes from 3284a and kept only conversion changes. The removed changes will go to 3284b. Rebased 3284a against trunk 11162.

Created new task branch 3284b and merged in all the changes from 3284a before the runtime changes were removed.

#23 - 08/25/2017 05:23 PM - Hynek Cihlar

Greg Shah wrote:

> Code Review Task branch 3284a Revision 11185
>
> I'm fine with the changes.
>
> One question: Shouldn't the gap marking for window-state runtime remain "partial"? It was previously completely implemented except for delayed minimize.

Yes, fixed.

#24 - 08/25/2017 05:24 PM - Hynek Cihlar

3284a conversion test passed.

#25 - 08/25/2017 05:25 PM - Greg Shah

Great! Please merge to trunk.

#26 - 08/25/2017 05:42 PM - Hynek Cihlar

3284a merged to trunk as revision 11163 and archived.

#27 - 08/25/2017 05:52 PM - Hynek Cihlar

Rebased 3284b against trunk 11163.

#28 - 09/08/2017 08:43 AM - Greg Shah

Please provide some details of the issues related to ALWAYS-ON-TOP.

#29 - 09/08/2017 08:51 AM - Hynek Cihlar

Greg Shah wrote:

> Please provide some details of the issues related to ALWAYS-ON-TOP.

There are not really any issues there. The attribute introduces another level of window ordering complexity and quite a few other cases depending on the actual windows setup (regular vs frame dialogs vs system dialogs vs alert boxes and their relationships) and the respective values of TOP-ONLY and ALWAYS-ON-TOP.

A potential issue may be Swing and its window positioning limitations - the native 4GL behavior cannot be expressed natively in the Swing driver.

#30 - 09/08/2017 08:57 AM - Greg Shah

> A potential issue may be Swing and its window positioning limitations - the native 4GL behavior cannot be expressed natively in the Swing driver.

OK. Document any limitations here and we can discuss. As with other items in the Swing driver, we can probably live with some small limitations.

#31 - 09/11/2017 09:43 AM - Hynek Cihlar

Hynek,

OK. Please report this in 3284.

This branch will merge to trunk after 3330a and 3222a.

Thanks,
Greg

On 09/11/2017 08:58 AM, Hynek Cihlar wrote:

> Greg,
>
> except of the Windows8 and Windows10 small close window caption icons I will have the changes ready for regression testing today. Some time will be required for regression testing and if nothing major will come up, the regression tested implementation will be ready sometime tomorrow.

> As of the small close window caption button icons. I don't have a simple answer, it will require some time to play with native Win32 to replicate the window appearance (assuming Progress doesn't draw the window title itself) and then to capture the button images. I think this will be ready yet during tomorrow or on Wednesday.
>
> Thanks,
> Hynek

#32 - 09/11/2017 10:03 AM - Hynek Cihlar

With trunk revision 11145 (among the other benefits) we lost the flexibility of arbitrary stacking top-level windows in Swing GUI driver. This makes it impossible to fully support the runtime behavior of WINDOW:TOP-ONLY and WINDOW:ALWAYS-ON-TOP attributes in Swing.

The closest I got was to use the native Swing's always-on-top feature. When WINDOW:TOP-ONLY or WINDOW:ALWAYS-ON-TOP is set the driver will call java.awt.Window.setAlwaysOnTop() on the respective window. The method to some extent depends on the underlying windowing system and so the exact behavior may differ from platform to platform.

#33 - 09/11/2017 11:47 AM - Greg Shah

> This makes it impossible to fully support the runtime behavior of WINDOW:TOP-ONLY and WINDOW:ALWAYS-ON-TOP attributes in Swing.

I assume you can match the behavior exactly in the web client virtual desktop mode. Correct? Web client embedded mode has no matching concept.

Please list the specific deviations that will be seen in the Swing client.

#34 - 09/11/2017 12:19 PM - Hynek Cihlar

Greg Shah wrote:

> I assume you can match the behavior exactly in the web client virtual desktop mode. Correct? Web client embedded mode has no matching concept.

This is correct.

> Please list the specific deviations that will be seen in the Swing client.

The Swing deviations:

- Active window with TOP-ONLY set will appear on top of window with ALWAYS-ON-TOP set.
- Window with TOP-ONLY set will be always on top of other system windows (the exact behavior is platform specific).
- Modal window (dialog-boxes, alert-boxes and system dialogs) with its parent having TOP-ONLY set will appear on top of windows with

ALWAYS-ON-TOP set.
- Modal window (dialog-boxes, alert-boxes and system dialogs) with its parent with unset TOP-ONLY and ALWAYS-ON-TOP will appear on top of windows with TOP-ONLY or ALWAYS-ON-TOP set.
- Setting TOP-ONLY on a visible window will move the window to the very top of z-order regardless of other windows with ALWAYS-ON-TOP set (platform specific).
- Unsetting TOP-ONLY or ALWAYS-ON-TOP on a visible window will not change the Window's z-order (platform specific).

#35 - 09/12/2017 01:51 PM - Hynek Cihlar

FYI, I had to revert the changes from trunk revision 11145 in Swing GUI driver related to window activation. The revision assumed that all window activations come from the system. But there are cases when a window must be activated on window close or when window minimized. In this case it is up to the FWD client to select the next window to activate and issue the request to the driver.

#36 - 09/14/2017 08:26 AM - Hynek Cihlar

3284b revision 11173 passed ChUI regression testing and GUI regression testing with one condition. In Web GUI when overlay window is opened (combo box dropdown) and an action is performed that directly opens another modal window, the modal window is closed right away. This is caused by a latent bug in Web GUI, it doesn't send window activated event for the overlay owner.

Since the issue above is localized in the js sources, the revision can be already reviewed.

Still a todo is the small close window caption button for windows 10 and 8.

#37 - 09/14/2017 04:17 PM - Hynek Cihlar

Hynek Cihlar wrote:

> In Web GUI when overlay window is opened (combo box dropdown) and an action is performed that directly opens another modal window, the modal window is closed right away. This is caused by a latent bug in Web GUI, it doesn't send window activated event for the overlay owner.

This is fixed in 3284b revision 11174.

#38 - 09/15/2017 03:14 PM - Constantin Asofiei

Hynek, review for 3284b rev 11174:

1. have you tested how window-delayed-minimize and ALWAYS-ON-TOP work with parent-child relationships for windows?
2. Windowwidget.setAlwaysOnTop - you've removed the 'window is realized' check - is this OK?
3. AbstractFileChooserDialog, AlertBoxGuiImpl - have no functional changes
4. OutputManager - please merge the history entries
5. TopLevelWindow:852 - repaint(); // TODO: repaint whole window (?) we need only the window border/titlebar - can TopLevelWindow.repaintDecorations be used?
6. SwingGuiDriver.setAlwaysOnTop - missing javadoc
7. GuiWebSocket.createWindow - missing javadoc for showInTaskbar field
8. please add a WindowWidget.setDisableRedraw(logical), too
9. WindowManager.delayedMinimizeWindows - please make this context-local

#39 - 09/18/2017 05:21 AM - Hynek Cihlar

Constantin Asofiei wrote:

> Hynek, review for 3284b rev 11174:
>
> 1. have you tested how window-delayed-minimize and ALWAYS-ON-TOP work with parent-child relationships for windows?

Yes.

> 1. Windowwidget.setAlwaysOnTop - you've removed the 'window is realized' check - is this OK?

Yes, ALWAYS-ON-TOP can be changed when window is realized.

> 1. TopLevelWindow:852 - repaint(); // TODO: repaint whole window (?) we need only the window border/titlebar - can TopLevelWindow.repaintDecorations be used?

The TODO already exists in trunk. Can TopLevelWindow.repaintDecorations be used? I'm not sure, I will check.

#40 - 09/27/2017 06:12 PM - Hynek Cihlar

3284b revision 11184 resolves all the points from the last review, resolves the layout and draw of window small close caption button in Windows 8 and Windows 10 and fixes several regressions. Currently there are no open points left. ChUI and GUI regression testing after the rebase and the latest changes is yet in progress.

3284b is rebased against trunk 11170.

#41 - 09/28/2017 11:32 AM - Constantin Asofiei

Hynek, I'm OK with the change in 3284b revision 11184. Can you please do some tests with the large GUI app?

#42 - 09/28/2017 11:40 AM - Hynek Cihlar

Constantin Asofiei wrote:

> Can you please do some tests with the large GUI app?

Yes, I am currently testing Hotel GUI.

#43 - 09/28/2017 11:57 AM - Greg Shah

I think he means the customer GUI app in which we had 10 documented scenarios that can be used for testing.

#44 - 09/29/2017 04:11 AM - Hynek Cihlar

The last ChUI regression test round finished with failures, I am not yet sure whether these were caused by the changes in the task branch, I am not able to replicate the failures. I started another run.

#45 - 09/29/2017 06:53 PM - Hynek Cihlar

3284b revision 11186 fixes a sort of regression that influences how the client interacts with test harness. The problem was with the default window not accepting input right away after the window was shown. The problem could not be spotted when manually tested, only in the harness.

The task branch is rebased against trunk revision 11171 and passed GUI regression testing. A ChUI regression run is in progress, when it passes it will be ready for merge to trunk.

#46 - 09/30/2017 02:54 PM - Hynek Cihlar

The last regression test run finished with some negative failures. I started a new run.

#47 - 10/01/2017 07:18 PM - Hynek Cihlar

3284b passed ChUI regression tests. Please review.

#48 - 10/02/2017 07:47 AM - Constantin Asofiei

Hynek Cihlar wrote:

> 3284b passed ChUI regression tests. Please review.

I'm OK with the changes in 3284b rev 11187.

#49 - 10/02/2017 08:07 AM - Greg Shah

Please merge 3284b to trunk.

#50 - 10/02/2017 08:17 AM - Hynek Cihlar

3284b merged to trunk as revision 11172 and archived.

#51 - 10/02/2017 08:45 AM - Greg Shah

- Assignee set to Hynek Cihlar

- Status changed from New to Closed

- % Done changed from 0 to 100