

User Interface - Bug #2557

Bug # 2677 (New): fix drawing and functional differences between P2J GUI and 4GL GUI

Fix MENUBAR layouting in a window

04/27/2015 12:43 PM - Vadim Gindin

Status: New	Start date:
Priority: Low	Due date:
Assignee: Vadim Gindin	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version: Cleanup and Stabilization for GUI	case_num:
billable: No	
vendor_id: GCD	
Description	
Related issues:	
Related to User Interface - Feature #1790: implement menu support	Closed
Related to User Interface - Bug #3553: ChUI menus don't work	New

History

#1 - 04/27/2015 01:09 PM - Vadim Gindin

Current MENUBAR implementation has several layout problems, that should be solved in this task.

1. If source procedure outputs some string to message area it being hidden when MENUBAR is drawn right after start.
Here is testing procedure:

```
def var r as int init 1.
def var i as int init 1.

def var outp as char extent 24.

def sub-menu sbm
  menu-item mii label "iii".

def menu m menubar
  menu-item qqq label "qqq".
  sub-menu sbm label "Orwell".
  menu-item print label "print".

on choose of menu-item print do:
  i = 1.
  repeat while i < r:
    put screen row i col 22 string(outp[i], "x(60)").
    i = i + 1.
  end.
end.

on menu-drop of sub-menu sbm do:
  message r "menu-drop".
  def var ch as char.
  ch = last-event:label + " " + string(keycode(last-event:label)) + " " + last-event:event-type + " " +
    string(last-event:code) + " " + last-event:function + " " + last-event:type + " " + string(LASTKEY)
  .
  outp[r] = ch.
  r = r + 1.
end.

message "bogus message so that the message area is not empty".
assign current-window:menubar = menu m:handle.
wait-for choose of menu-item qqq.
```

This procedure will be referenced in other bugs below.

Right during start the message "bogus message so that the message area is not empty" is drawn and than become hidden when MENUBAR is drawn. For contentPane of the window (that is an Container instance) the method AbstractContainer.getDrawableWidgets() returns the list with only MENUBAR, when it was added. From the other side first time this contentPane's method is called it contains 2 message lines and 1 status line.

2. When such message as "bogus ..." is called (see previous testing procedure), and the user tries to navigate sub-menu "Orwell", the first menu-item from that menu does not gain default focus first time. It happens because of the message pause being executed. Implement the same behavior. Take note that MENU-DROP triggers are not intended to contain I/O commands (see "Language Reference" and referenced task in the note) and we only have to support MENU-DROP triggers behavior only in such cases. See testing procedure for example of MENU-DROP event that does not use I/O operations.

3. When window contain MENUBAR there are some specific behavior is met:

- If the window already contains some frames they are shifted down and those heights are readjusted.
See Constantin's comment in related task:

My suggestion is to finish the menu's drawing/event/etc first and work on this in a second phase, as some refactoring might needed (I think the frames in ChUI can't be parented to the window anymore, we need a "workspace" container as we use in GUI).

- When PUT SCREEN outputs some message to the first line and then MENUBAR is attached - it is drawn right over printed message without clearing.
- If some frame has focus MENUBAR can be accessed using "F3" key. I'm Recalling,that MENUBAR is not a tab order member and cannot be navigated using TAB.

See some tests in related tasks.

#2 - 09/08/2015 09:49 AM - Greg Shah

- Parent task set to #2677

We need to test to see if these issues are still a problem.

#3 - 10/12/2015 04:56 PM - Greg Shah

- Assignee set to Vadim Gindin

- Target version set to Milestone 12

- Start date deleted (04/27/2015)

#4 - 10/14/2015 02:23 PM - Greg Shah

Any needed changes should be put into 1811s.

#5 - 10/15/2015 04:25 PM - Vadim Gindin

I'm recalling. Issue 1. Progress shows "bogus .." message and does not show standard "Run-time error in WAIT-FOR.." message. P2J works vice

versa: shows "Run-time error in WAIT-FOR.." message and doesn't show "bogus .." message.

Have a look at 3 last procedure lines:

```
message "bogus message so that the message area is not empty".  
assign current-window:menubar = menu m:handle.  
wait-for choose of menu-item qqq.
```

Interesting findings.

1. P2J shows "bogus .." message in a window without menubar (menubar is not assigned yet - it is the next statement). PROGRESS shows bogus message in a window with menubar. Why? It looks like MESSAGE call is deferred as "pending" till some moment (WAIT-FOR ?). What moment it could be?
2. P2J shows "bogus .." message but overwrites it later with "Run-time error in WAIT-FOR..". Although the window does not contain frames at all PROGRESS does not show "Run-time error.." message (see TC.warnNoEnabledFrames). It looks like there is another criteria of warning show: an absence of pending messages. What do you think?

#6 - 10/16/2015 08:41 AM - Greg Shah

1. Is this problem in ChUI only or also in GUI?
2. You state this:

PROGRESS shows bogus message in a window with menubar. Why? It looks like MESSAGE call is deferred as "pending" till some moment (WAIT-FOR ?). What moment it could be?

This doesn't seem correct for ChUI. Try this:

```
message "bogus message so that the message area is not empty".  
pause. <--- ADD THIS  
assign current-window:menubar = menu m:handle.  
wait-for choose of menu-item qqq.
```

When the message is displayed, the menubar is not there yet.

3. You state this:

P2J shows "bogus .." message but overwrites it later with "Run-time error in WAIT-FOR..". Although the window does not contain frames at all PROGRESS does not show "Run-time error.." message (see TC.warnNoEnabledFrames).

When MENUBAR is assigned a valid menu, does the 4GL ever generate a "Run-time error in WAIT-FOR.." message?

If you remove the MESSAGE statement does the 4GL behave the same way as when the MESSAGE statement is present?

Greg Shah wrote:

1. Is this problem in ChUI only or also in GUI?

This problem happen in GUI.

2. You state this:

PROGRESS shows bogus message in a window with menubar. Why? It looks like MESSAGE call is defered as "pending" till some moment (WAIT-FOR ?). What moment it could be?

This doesn't seem correct for ChUI. Try this:

[...]

When the message is displayed, the menubar is not there yet.

3. You state this:

P2J shows "bogus .." message but overwrites it later with "Run-time error in WAIT-FOR.. ". Although the window does not contain frames atall PROGRESS does not show "Run-time error.." message (see TC.warnNoEnabledFrames).

When MENUBAR is assigned a valid menu, does the 4GL ever generate a "Run-time error in WAIT-FOR.. " message?

If you remove the MESSAGE statement does the 4GL behave the same way as when the MESSAGE statement is present?

Yes. Behaviour is the same.

#8 - 10/16/2015 11:36 AM - Greg Shah

Yes. Behaviour is the same.

So the MESSAGE statement has nothing to do with this problem?

What about when you put a pause after the message statement in GUI, I assume that (like ChUI) the menubar is not there?

Also, you didn't answer this question:

When MENUBAR is assigned a valid menu, does the 4GL ever generate a "Run-time error in WAIT-FOR.." message?

#9 - 10/16/2015 02:36 PM - Vadim Gindin

Greg Shah wrote:

Yes. Behaviour is the same.

So the MESSAGE statement has nothing to do with this problem?

I beg your pardon, I probably wasn't enough clear. PROGRESS shows "Runtime error in WAIT-FOR..." warning in cases when window does not contains enabled frames and contains a MENUBAR. The "Absence of enabled frames" is a founded criteria of that warning show in PROGRESS. It was implemented in TC.warnNoEnabledFrames() that is called from TC.waitFor(..).

MESSAGE statement before WAIT-FOR is just another case, when message area will not be empty at the moment of "Runtime error in WAIT-FOR..." warning. That case shown the difference in behaviour between PROGRESS and P2J: PROGRESS shows MESSAGE string instead of warning but P2J vice versa.

What about when you put a pause after the message statement in GUI, I assume that (like ChUI) the menubar is not there?

Yes, you are right. MENUBAR is not shown.

Also, you didn't answer this question:

When MENUBAR is assigned a valid menu, does the 4GL ever generate a "Run-time error in WAIT-FOR.." message?

The warning only denotes the absence of enabled frame in a window is menubar assigned. It doesn't matter if MENUBAR is valid menu or not valid.

#10 - 10/16/2015 02:50 PM - Greg Shah

I beg your pardon, I probably wasn't enough clear. PROGRESS shows "Runtime error in WAIT-FOR..." warning in cases when window does not contains enabled frames and contains a MENUBAR. The "Absence of enabled frames" is a founded criteria of that warning show in PROGRESS. It was implemented in TC.warnNoEnabledFrames() that is called from TC.waitFor(..).

MESSAGE statement before WAIT-FOR is just another case, when message area will not be empty at the moment of "Runtime error in WAIT-FOR..." warning. That case shown the difference in behaviour between PROGRESS and P2J: PROGRESS shows MESSAGE string instead of warning but P2J vice versa.

If you remove the MESSAGE statement from this testcase, then will Progress generate the "Runtime error in WAIT-FOR..." warning?

#11 - 10/16/2015 03:13 PM - Vadim Gindin

Greg Shah wrote:

If you remove the MESSAGE statement from this testcase, then will Progress generate the "Runtime error in WAIT-FOR..." warning?

Yes.

#12 - 10/17/2015 10:59 AM - Greg Shah

Are there any other issues listed above that are still a problem?

#13 - 10/19/2015 10:39 AM - Vadim Gindin

F3 focusing the menubar (fixed)

Open issues:

1. Hiding bogus message.
2. Focusing is still a problem.
3. PUT SCREEN cleaning.
4. Adjusting heights (needed to check).

#14 - 10/20/2015 07:08 AM - Vadim Gindin

I'm working on heights adjustment of window content when it contains MENUBAR (including dynamic attachment). When window is fully filled (all rows), then when menubar is attached - all content is shifted down and displaces status string. How to correctly set contentPane size? does the content pane height include message area and status line heights or something else?

#15 - 10/20/2015 07:10 AM - Constantin Asofiei

Vadim Gindin wrote:

I'm working on heights adjustment of window content when it contains MENUBAR (including dynamic attachment). When window is fully filled (all rows), then when menubar is attached - all content is shifted down and displaces status string. How to correctly set contentPane size? does the content pane height include message area and status line heights or something else?

I think you need to trigger a window resize (for the workspace, not full window): set the workspace height to the workspace.height - menu.height and resize the window. Same when the menu is detached, but this time increase the height.

See the WindowGuiImpl.resizeWorkspace and WindowGuiImpl.resizeTo. This will take care of re-laying out the window.

#16 - 10/20/2015 07:40 AM - Vadim Gindin

I forgot to add that my question is about CHUI.

#17 - 10/20/2015 07:50 AM - Vadim Gindin

I've set contentPane's height to workspace height minus menubar height. Now status line is not displaced but the content height was not reduced.

#18 - 10/20/2015 08:45 AM - Constantin Asofiei

Vadim Gindin wrote:

I forgot to add that my question is about CHUI.

ChUI is tricky - as I recall, menubar and all widgets are placed in the same container in P2J, correct? Please post the widget hierarchy from WidgetBrowser (make sure you have a menubar added).

Same as for GUI, after the menubar is placed in 4GL, in ChUI the frames can be resized (sometimes frames become scrollable, other times - I don't recall the exact case - the frame can extend further, over the message area). Also, because menubar and frames are siblings, we will have problems when placing them explicitly via location attributes, if there is an active menubar. I wonder if is not the time to refactor a little the widget hierarchy and create a workspace equivalent in ChUI, where to place the frames...

Please also check something else: use PUT SCREEN to write something on the first row - what happens with this text, after menubar is added? Does it get pushed down or is overwritten by the menubar? Also, what happens with PUT SCREEN when menubar is active - does it start counting rows from below the menubar (I mean, PUT SCREEN on row 1 will be on menubar or below it?)

Are there any other issues in GUI, related to menubar?

#19 - 10/20/2015 09:10 AM - Greg Shah

Vadim: this ChUI work can wait. Document your findings and any code changes that are not safe to check in.

Please list all open GUI issues for this task.

#20 - 10/20/2015 09:17 AM - Vadim Gindin

Constantin Asofiei wrote:

Vadim Gindin wrote:

I forgot to add that my question is about CHUI.

ChUI is tricky - as I recall, menubar and all widgets are placed in the same container in P2J, correct? Please post the widget hierarchy from WidgetBrowser (make sure you have a menubar added).

```
WindowChuiImpl
  BorderedPanelImpl
    MessageLine (?)
    FrameChuiImpl
    FrameChuiImpl
    MessageLine
    MessageLine
    MenuChuiImpl
    StatusLineImpl
```

Same as for GUI, after the menubar is placed in 4GL, in ChUI the frames can be resized (sometimes frames become scrollable, other times - I don't recall the exact case - the frame can extend further, over the message area). Also, because menubar and frames are siblings, we will have problems when placing them explicitly via location attributes, if there is an active menubar. I wonder if is not the time to refactor a little the widget hierarchy and create a workspace equivalent in ChUI, where to place the frames...

Yes, probably some refactoring will be needed there.

Please also check something else: use PUT SCREEN to write something on the first row - what happens with this text, after menubar is added? Does it get pushed down or is overwritten by the menubar? Also, what happens with PUT SCREEN when menubar is active - does it start counting rows from below the menubar (I mean, PUT SCREEN on row 1 will be on menubar or below it?)

PUT SCREEN overwrites MENUBAR (is printed over) and vice versa: MENUBAR overwrites the string printed by PUT SCREEN. More, when MENUBAR that was printed over some string (printed by PUT SCREEN before) and then detached the online space under MENUBAR items is cleared. The spaces between, before and after continuing to hold characters printed by PUT SCREEN:

```
00000000000000000000000000000000 (PUT SCREEN)
Exit0Orwell1000000000000000000000 (attached MENUBAR)
  0      00000000000000000000000000 (detached PROGRESS)
Exit0Orwell1000000000000000000000 (detached P2J)
```

Are there any other issues in GUI, related to menubar?

I'll check it today a little later if you don't mind.

#21 - 10/20/2015 03:32 PM - Vadim Gindin

For GUI there is only one issue - "hiding bogus message":

PROGRESS shows "Runtime error in WAIT-FOR..." warning in cases when window does not contains enabled frames and contains a MENUBAR. The "Absence of enabled frames" is a founded criteria of that warning show in PROGRESS. It was implemented in TC.warnNoEnabledFrames() that is called from TC.waitFor(..).

MESSAGE statement before WAIT-FOR is just another case, when message area will not be empty at the moment of "Runtime error in WAIT-FOR..." warning. That case shown the difference in behaviour between PROGRESS and P2J: PROGRESS shows MESSAGE string instead of warning but P2J vice versa.

The new issue: when MENUBAR dynamically is assigned (from button's trigger) menu items are not displayed.

#22 - 10/20/2015 03:35 PM - Greg Shah

The new issue: when MENUBAR dynamically is assigned (from button's trigger) menu items are not displayed.

This one is only in ChUI?

#23 - 10/20/2015 03:36 PM - Vadim Gindin

This is only for GUI. It just wasn't documented in this task earlier.

#24 - 10/20/2015 03:50 PM - Greg Shah

- *Priority changed from Normal to Low*

OK, please pause work on this task.

#25 - 03/23/2016 05:18 PM - Greg Shah

- *Target version changed from Milestone 12 to Milestone 16*

#26 - 03/29/2016 11:07 AM - Vadim Gindin

Current branch is 2557a (revno 10987). It is rebased with the current trunk revision and contains dynamic window assignment fix.

I recall that the message Run-time error in WAIT-FOR at line <line_num> (relative to expanded source) of <program_name>. (3269) appears in GUI message area at the start of any procedure that has the window with assigned menubar but without enabled frames.

I've found that after some moment the message is incorrectly split to a several lines with the width approximately 21. This error also exist in trunk. Is it

a known bug?

#27 - 03/29/2016 12:40 PM - Vadim Gindin

I'm sorry. It was false alarm about incorrect splitting. It was related somehow with my changes. I've fixed it.

I've also found that WARNING noted earlier is not shown if window is dynamic. I've also fixed it. During my solution I've added new flag to `ThinClient.noEnabledFramesCheckPassed` denoting exactly passing of corresponding check during WAIT-FOR processing.

This flag is used in `Window.afterConfigUpdate(..)` for the cases when menubar is not assigned to a window at procedure start, but it is assigned in button trigger. Using that flag we can set menubar visible if it wasn't done in `noEnabledFramesCheck` in WAIT-FOR check.

Branch revision is 10988.
Needed review.

#28 - 03/29/2016 01:51 PM - Constantin Asofiei

Review for 2557a rev 10988:

1. `MessageAreaGuiImpl`:
 - you have an unneeded `import java.util.Arrays`;
 - is missing history entry
2. `Window`:
 - you have a `getContentPane().remove(menubar);`. But I think is not enough. You should call `menubar.destroy()` and also de-register it (and all its sub-components - submenus, menuitems) from the registry - if they are added (I don't recall if they are kept in the registry or not).
3. `ThinClient`:
 - I understand the reason for `noEnabledFramesCheckPassed` flag, but I'm not really fond of the name... how about a flag which tells the window if it can attach the menubar or not? Something like `canAttachMenubar`? Is more explanatory.

#29 - 03/29/2016 11:29 PM - Vadim Gindin

Thanks. I'll fix it.

There are no more GUI issues, should I fix CHUI in this task?

#30 - 03/30/2016 02:05 AM - Vadim Gindin

Constantin Asofiei wrote:

Review for 2557a rev 10988:

..

1. `Window`:
 - you have a `getContentPane().remove(menubar);`. But I think is not enough. You should call `menubar.destroy()` and also de-register it (and all its sub-components - submenus, menuitems) from the registry - if they are added (I don't recall if they are kept in the registry or not).

Yes, menu widgets are registered in `WidgetRegistry`, but it happen during MENU creation stage during pushing static menu description to the client `ThinClient.pushMenuDescription(..)`: i.e. not in attachment stage. If I'll deregister MENU widgets sub-tree in `Window.afterConfigUpdate` I'll not be able

to attach menubar there again during the next action in this method.

1. ThinClient:

- I understand the reason for noEnabledFramesCheckPassed flag, but I'm not really fond of the name... how about a flag which tells the window if it can attach the menubar or not? Something like canAttachMenubar? Is more explanatory.

OK. It is more about visibility.. May be isMenubarShowNeeded?

#31 - 03/30/2016 02:40 AM - Constantin Asofiei

Vadim Gindin wrote:

Constantin Asofiei wrote:

Review for 2557a rev 10988:

..

1. Window:

- you have a getContentPane().remove(menubar);. But I think is not enough. You should call menubar.destroy() and also de-register it (and all its sub-components - submenus, menuitems) from the registry - if they are added (I don't recall if they are kept in the registry or not).

Yes, menu widgets are registered in WidgetRegistry, but it happen during MENU creation stage during pushing static menu description to the client ThinClient.pushMenuDescription(..): i.e. not in attachment stage. If I'll deregister MENU widgets sub-tree in Window.afterConfigUpdate I'll not be able to attach menubar there again during the next action in this method.

Ah, OK, I forgot it can be reused. So the code is OK.

1. ThinClient:

- I understand the reason for noEnabledFramesCheckPassed flag, but I'm not really fond of the name... how about a flag which tells the window if it can attach the menubar or not? Something like canAttachMenubar? Is more explanatory.

OK. It is more about visibility.. May be isMenubarShowNeeded?

Yes, isMenubarShowNeeded is OK, but rename the ThinClient.noEnabledFramesCheckPassed too.

#32 - 03/30/2016 04:19 AM - Vadim Gindin

I've fixed remarks and rebased the branch. Current revision is 10990. Can I run regression testing? Should I work on CHUI part of this task here?

#33 - 03/30/2016 04:27 AM - Constantin Asofiei

There are two javadoc errors in the code:

```
[javadoc] /home/ca/testing/branches/2557a/src/com/goldencode/p2j/ui/chui/ThinClient.java:2638: warning - Tag
@link: can't find afterConfigUpdate() in com.goldencode.p2j.ui.client.Window
[javadoc] /home/ca/testing/branches/2557a/src/com/goldencode/p2j/ui/chui/ThinClient.java:19178: warning - Ta
g @link: reference not found: menubarShowNotNeeded
```

Otherwise, you can go ahead and test. The ChUI part I think will be in another task/branch.

#34 - 03/30/2016 08:46 AM - Greg Shah

Please create a new task in the UI project with all the details on the ChUI issue. It should include the exact code needed to recreate the problem and a full list of the issues that need resolution. We will not work that problem now.

#35 - 04/06/2016 04:59 AM - Vadim Gindin

I've rebased the branch to current trunk revision. Current 2557a revision is 10998.

#36 - 07/26/2016 12:20 PM - Greg Shah

Vadim: did you ever create another task for the remaining ChUI items from this task?

#37 - 07/26/2016 12:43 PM - Vadim Gindin

Greg Shah wrote:

Vadim: did you ever create another task for the remaining ChUI items from this task?

It seems I didn't.. There is [#3134](#) task, also containing CHUI menu bugs. May be I should add these bugs to that task?

#38 - 07/26/2016 06:20 PM - Greg Shah

May be I should add these bugs to that task?

No. Please create a separate task and link to it here as a related task.

#39 - 07/27/2016 01:29 AM - Vadim Gindin

Greg Shah wrote:

May be I should add these bugs to that task?

No. Please create a separate task and link to it here as a related task.

I've created the new task [#3157](#) task for CHUI bugs.

#40 - 11/07/2016 04:32 AM - Vadim Gindin

Vadim Gindin wrote:

Greg Shah wrote:

May be I should add these bugs to that task?

No. Please create a separate task and link to it here as a related task.

I've created the new task [#3157](#) task for CHUI bugs.

I've rebased the branch 2557a to current trunk revision. Now branch revision is 11131. Please review.

#41 - 11/07/2016 09:01 AM - Greg Shah

Please summarize the purpose of the changes in 2557a.

Hynek: please do a code review.

#42 - 11/07/2016 09:50 AM - Hynek Cihlar

Code review 2557a.

MessageAreaGuilImpl.java is missing a history entry.

Window.afterConfigUpdate()

I think there is a potential NPE (which was there before 2557a) in Window.afterConfigUpdate() at the line menubar.setVisible(true) with no check for menubar being null, while there is a check for null a several lines above.

Vadim, please post a sample that demonstrates the menubar visibility issue. I'll need it to finish the code review.

#43 - 11/07/2016 10:45 AM - Vadim Gindin

/menu/gui/mbar_warn.p

```
define sub-menu topic
  menu-item numbr label "Number"
  menu-item addr label "Address".

define sub-menu move
  menu-item forward label "Forward"
  menu-item backward label "Back"
  rule
  menu-item toggle label "Toggle" TOGGLE-BOX.

define sub-menu quitit
  menu-item quititem label "E&xit".

define menu mbar menubar
  sub-menu topic label "Topic"
  sub-menu move label "Move"
  sub-menu quitit label "E&xit".

on choose of menu-item numbr
  display "Number".

on choose of menu-item addr
  display "Address".

on choose of menu-item forward
  display "forward".

on choose of menu-item backward
  display "backward".

assign current-window:menubar = menu mbar:handle.

message "SOME TEXT".

wait-for choose of menu-item quititem.
```

#44 - 11/07/2016 03:34 PM - Hynek Cihlar

Vadim Gindin wrote:

```
/menu/gui/mbar_warn.p
```

```
[...]
```

Vadim, send please also the steps to reproduce the issue.

#45 - 11/07/2016 11:51 PM - Vadim Gindin

Hynek Cihlar wrote:

Vadim Gindin wrote:

```
/menu/gui/mbar_warn.p
```

```
[...]
```

Vadim, send please also the steps to reproduce the issue.

Yes, sorry, here is a description of the change. This change relates to the following WARNING: Run-time error in WAIT-FOR at line <line> (relative to expanded source) of <program_name>. (3269). This warning is often met in GUI menu test cases right after start in MESSAGE-AREA, when a test contains MENUBAR and no enabled frames. You can see implementation in ThinClient.warnNoEnabledFrames()

This change fixes some border case of WARNING problem: When MESSAGE-AREA is not empty, then we must not show the WARNING.

The test case prints message "SOME TEXT" to MESSAGE-AREA before WAIT-FOR. Right after start we must see only that message in MESSAGE-AREA, but in trunk we see this message and noted WARING below. I.e. just run the test and look at MESSAGE-AREA.

#46 - 11/08/2016 07:00 AM - Hynek Cihlar

Vadim Gindin wrote:

Hynek Cihlar wrote:

Vadim Gindin wrote:

/menu/gui/mbar_warn.p

[...]

Vadim, send please also the steps to reproduce the issue.

Yes, sorry, here is a description of the change. This change relates to the following WARNING: Run-time error in WAIT-FOR at line <line> (relative to expanded source) of <program_name>. (3269). This warning is often met in GUI menu test cases right after start in MESSAGE-AREA, when a test contains MENUBAR and no enabled frames. You can see implementation in ThinClient.warnNoEnabledFrames()

This change fixes some border case of WARNING problem: When MESSAGE-AREA is not empty, then we must not show the WARNING.

The test case prints message "SOME TEXT" to MESSAGE-AREA before WAIT-FOR. Right after start we must see only that message in MESSAGE-AREA, but in trunk we see this message and noted WARING below. I.e. just run the test and look at MESSAGE-AREA.

How does this relate to the new method ThinClient.isMenubarShowNeeded()? What would be the test case for it?

#47 - 11/08/2016 07:48 AM - Hynek Cihlar

Hynek Cihlar wrote:

How does this relate to the new method ThinClient.isMenubarShowNeeded()? What would be the test case for it?

In other words, what would be the test case to test out the execution path of ThinClient.isMenubarShowNeeded()?

Hynek Cihlar wrote:

Hynek Cihlar wrote:

How does this relate to the new method `ThinClient.isMenubarShowNeeded()`? What would be the test case for it?

In other words, what would be the test case to test out the execution path of `ThinClient.isMenubarShowNeeded()`?

Oh, now I've found. The testcase is `menu/menubar.p`:

```
def button b1 label "Hide menubar".
def button b2 label "Show menubar".
def button b3 label "Error".
def button ext label "Exit".

def frame f b1 skip b2 skip b3 skip ext.

enable all with frame f.

def sub-menu sm
  menu-item m1 label "&&&&Toyota&&".
  menu-item m2 label "&Dodge".
  menu-item m3 label "&&Chrysler".

def sub-menu sm0
  menu-item m8 label "&Re&nault&".
  menu-item test label "T&est&".

define menu m menubar.
  sub-menu sm label "sm".
  sub-menu sm0 label "sm0".

on choose of menu-item m1
  display "toyotaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa".

on choose of b2
  assign current-window:menubar = menu m:handle.

on choose of b1
  assign current-window:menubar = ?.

on choose of b3
  menu m:visible = false.

wait-for choose of ext.
```

Scenario is simple. Press "Show Menubar" it must be shown and press "Hide Menubar" and it must be hidden.

#49 - 11/08/2016 04:11 PM - Hynek Cihlar

Vadim Gindin wrote:

Hynek Cihlar wrote:

Hynek Cihlar wrote:

How does this relate to the new method `ThinClient.isMenubarShowNeeded()`? What would be the test case for it?

In other words, what would be the test case to test out the execution path of `ThinClient.isMenubarShowNeeded()`?

Oh, now I've found. The testcase is `menu/menubar.p`:
[...]

Scenario is simple. Press "Show Menubar" it must be shown and press "Hide Menubar" and it must be hidden.

Please help me understand the logic behind the addition of `ThinClient.isMenubarShowNeeded()`. Under what conditions the menu bar should not be shown when assigned to the window?

Also the implementation will fail when the menu bar is assigned unconditionally before the `WAIT-FOR` statement. Please see the modified test case:

```
def button b1 label "Hide menubar".
def button b2 label "Show menubar".
def button b3 label "Error".
def button ext label "Exit".

def frame f b1 skip b2 skip b3 skip ext.

enable all with frame f.

def sub-menu sm
  menu-item m1 label "&&&&Toyota&&".
  menu-item m2 label "&Dodge".
  menu-item m3 label "&&Chrysler".

def sub-menu sm0
  menu-item m8 label "&Re&nault&".
  menu-item test label "T&est&".

define menu m menubar.
  sub-menu sm label "sm".
  sub-menu sm0 label "sm0".

on choose of menu-item m1
  display "toyotaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa".

on choose of b2
  assign current-window:menubar = menu m:handle.

on choose of b1
  assign current-window:menubar = ?.

on choose of b3
  menu m:visible = false.

/* this assignment was added */
current-window:menubar = menu m:handle.

wait-for choose of ext.
```

To reproduce the issue, press "Hide menubar" and then "Show menubar" buttons, the result will be an empty menu bar space in the current window.

#50 - 11/08/2016 04:21 PM - Hynek Cihlar

Vadim Gindin wrote:

```
/menu/gui/mbar_warn.p
```

```
[...]
```

Btw., the test case abends when a sub-menu is selected from the menu-bar by mouse and the mouse pointer is moved to the adjacent sub-menu. Can be reproduced in current trunk.

Greg, do you want this to be fixed as part of this issue?

#51 - 11/09/2016 12:47 AM - Vadim Gindin

Hynek Cihlar wrote:

Vadim Gindin wrote:

```
/menu/gui/mbar_warn.p
```

```
[...]
```

Btw., the test case abends when a sub-menu is selected from the menu-bar by mouse and the mouse pointer is moved to the adjacent sub-menu. Can be reproduced in current trunk.

Greg, do you want this to be fixed as part of this issue?

Hynek, I've rebased the branch and fixed this abend. It become simple after the branch 2966a was merged to trunk.

#52 - 11/09/2016 01:21 AM - Vadim Gindin

Hynek, the logic is approximately the following.

1. MENUBAR is hidden by default because of necessity to show WARNING if there are no enabled frames. We show the WARNING and after user presses some button - only than we show the MENUBAR.
2. The WARNING is show in WAIT-FOR processing.
3. Dynamic case (menubar.p). We don't need to show the WARNING. So for dynamic case we must show MENUBAR at once in Window.afterConfigUpdate(). This method is called several times in different stages of starting. So we need additional flag menubarShowNeeded denoting exactly dynamic case and possibility to show MENUBAR right away and not to wait WAIT-FOR processing.

#53 - 11/09/2016 02:36 AM - Vadim Gindin

Hynek, about the bug you found. I've debugged it and found that it is caused by the following.

When the MENUBAR is hidden Menu.destroy() is called:

```
public void destroy()
{
    screen().getRegistry().removeWidgetsRecursive((Widget) this);
    super.destroy();
}
```

When we remove Menu object from the registry, all it's children are destroyed. When we are trying to make Menu visible again, they are not restored in WidgetRegistry.pushMenuDescription() later:

```
..
    if (menu != null)
    {
        addWidgetsToMenu(menuId, menuDescr);
    }
..
```

because of menu is null.

By the way, SubMenu does not remove it's children and therefore restored successfully.

As I can see you added this code at some moment. Why it is needed to remove MENU object from widget registry?

#54 - 11/09/2016 05:36 AM - Hynek Cihlar

Vadim Gindin wrote:

Hynek, the logic is approximately the following.

1. MENUBAR is hidden by default because of necessity to show WARNING if there are no enabled frames. We show the WARNING and after user presses some button - only than we show the MENUBAR.

I have tried to show up the warning message on windev01 but failed, can you please point me to an example that demonstrates the warning message?

#55 - 11/09/2016 05:39 AM - Hynek Cihlar

Vadim Gindin wrote:

Hynek, about the bug you found. I've debugged it and found that it is caused by the following.

When the MENUBAR is hidden Menu.destory() is called:
[...]

If the menu should be made only invisible then I think destroy() should not be called. The expected semantic of destroy() is to cleanup any external resources (like the widget registrations in this case) and to recursively destroy the child widgets.

As I can see you added this code at some moment. Why it is needed to remove MENU object from widget registry?

The code was added as part of a larger mostly refactoring change, and I am not sure whether this is just refactored or new logic.

#56 - 11/09/2016 08:36 AM - Hynek Cihlar

- File Screenshot at 2016-11-09 14_35_32.png added

Hynek Cihlar wrote:

Vadim Gindin wrote:

Hynek, the logic is approximately the following.

1. MENUBAR is hidden by default because of necessity to show WARNING if there are no enabled frames. We show the WARNING and after user presses some button - only than we show the MENUBAR.

I have tried to show up the warning message on windev01 but failed, can you please point me to an example that demonstrates the warning message?

According to my short testing on windev01, the warning message (1) should be displayed as an error (see attachment) and (2) it doesn't seem to show up depending on any frame being enabled or not but on the fact whether the WAIT-FOR statement contains a menu item not assigned to a window (possibly the rule is more generic), see the code below.

```
define menu mbar menubar
  MENU-ITEM quitit label "E&xit".

/*assign current-window:menubar = menu mbar:handle.*/

DEF VAR i AS CHAR.
DEFINE FRAME f i.
ENABLE ALL WITH FRAME f.

wait-for choose of menu-item quitit.
```

#57 - 11/09/2016 08:51 AM - Vadim Gindin

- File *WARNING.png* added

Hynek Cihlar wrote:

Hynek Cihlar wrote:

Vadim Gindin wrote:

Hynek, the logic is approximately the following.

1. MENUBAR is hidden by default because of necessity to show WARNING if there are no enabled frames. We show the WARNING and after user presses some button - only than we show the MENUBAR.

I have tried to show up the warning message on windev01 but failed, can you please point me to an example that demonstrates the warning message?

According to my short testing on windev01, the warning message (1) should be displayed as an error (see attachment) and (2) it doesn't seem to show up depending on any frame being enabled or not but on the fact whether the WAIT-FOR statement contains a menu item not assigned to a window (possibly the rule is more generic), see the code below.

[...]

Hynek, you've found something other. Just open menu/gui/mbar_warn.p and comment the line message "SOME TEXT". When you run this test you will see the WARNING in message area, and you'll see the menubar only after you press some button. That's what I was talking about. See attached image.

#58 - 11/09/2016 09:03 AM - Vadim Gindin

According to my short testing on windev01, the warning message (1) should be displayed as an error (see attachment) and (2) it doesn't seem to show up depending on any frame being enabled or not but on the fact whether the WAIT-FOR statement contains a menu item not assigned to a window (possibly the rule is more generic), see the code below.

[...]

Hynek, you've found something other. Just open menu/gui/mbar_warn.p and comment the line message "SOME TEXT". When you run this test you will see the WARNING in message area, and you'll see the menubar only after you press some button. That's what I was talking about. See attached image.

Sorry, windev01 works very slowly for me last days.. Now I see. You're right about the WARNING rule: it probably more generic. Anyway current changes are only a small "border case".

P.S. Your attached image - looks different that what I got when I ran your test.

P.P.S. Do you insist that we should not remove Menu.destroy() method?

#59 - 11/09/2016 09:10 AM - Hynek Cihlar

Vadim Gindin wrote:

Hynek Cihlar wrote:

Hynek Cihlar wrote:

Vadim Gindin wrote:

Hynek, the logic is approximately the following.

1. MENUBAR is hidden by default because of necessity to show WARNING if there are no enabled frames. We show the WARNING and after user presses some button - only than we show the MENUBAR.

I have tried to show up the warning message on windev01 but failed, can you please point me to an example that demonstrates the warning message?

According to my short testing on windev01, the warning message (1) should be displayed as an error (see attachment) and (2) it doesn't seem to show up depending on any frame being enabled or not but on the fact whether the WAIT-FOR statement contains a menu item not assigned to a window (possibly the rule is more generic), see the code below.

[...]

Hynek, you've found something other. Just open menu/gui/mbar_warn.p and comment the line message "SOME TEXT". When you run this test you will see the WARNING in message area, and you'll see the menubar only after you press some button. That's what I was talking about. See attached image.

I do think the warning has nothing to do with the fact that a frame is enabled. Check out the following variation on the example above

```
define menu mbar menubar
  menu-item quitit label "E&xit".

CURRENT-WINDOW:VISIBLE = TRUE.
assign current-window:menubar = menu mbar:handle.

wait-for choose of menu-item quitit.
```

which doesn't produce any warning on windev01.

Btw., the warning is displayed as an error on windev01 when the 4GL code is executed from Procedure editor, this confused me.

#60 - 11/09/2016 09:12 AM - Hynek Cihlar

Vadim Gindin wrote:

P.P.S. Do you insist that we should not remove Menu.destroy() method?

IMHO the destroy() method should stay as-is but probably should not be called when the menu is being hidden to be made visible later on.

#61 - 11/09/2016 09:12 AM - Hynek Cihlar

Hynek Cihlar wrote:

Vadim Gindin wrote:

P.P.S. Do you insist that we should not remove Menu.destroy() method?

IMHO the destroy() method should stay as-is but probably should not be called when the menu is being hidden to be made visible later on.

Or making the menu visible again should reinitialize it completely.

#62 - 11/09/2016 12:51 PM - Vadim Gindin

Hynek Cihlar wrote:

Hynek Cihlar wrote:

Vadim Gindin wrote:

P.P.S. Do you insist that we should not remove Menu.destroy() method?

IMHO the destroy() method should stay as-is but probably should not be called when the menu is being hidden to be made visible later on.

Or making the menu visible again should reinitialize it completely.

1. destroy() is called from AbstractContainer.remove() when I call getContentPane().remove(menuubar). Are you sure that it is correct to call destroy from remove method?
2. In my testcase there is not just visibility, there is assignment current-window:menuubar = ?, so it would be incorrect to just change visibility in

implementation.

#63 - 11/09/2016 02:17 PM - Vadim Gindin

I've committed the MENU reinitialization fix and rebased the branch. Please review.

#64 - 11/09/2016 04:00 PM - Hynek Cihlar

Vadim Gindin wrote:

I've committed the MENU reinitialization fix and rebased the branch. Please review.

Code review 2557a.

MessageAreaGuilImpl.java is still missing a history entry.

MenuItemGuilImpl.java is missing history entry.

The above mentioned functional difference in handling the "Run-time error in WAIT-FOR at line.." error/warning message is still present.

#65 - 11/09/2016 04:34 PM - Greg Shah

Can someone please summarize the case that causes the warning? Does it only happen from the procedure editor?

#66 - 11/09/2016 04:50 PM - Hynek Cihlar

Greg Shah wrote:

Can someone please summarize the case that causes the warning?

One case where the warning is displayed is when WAIT-FOR contains a MENU-ITEM widget that is not "waitable", i.e. attached to an invisible window or window visible but the menu-item not attached to any window.

Frame states have no effect on the warning's appearance.

Does it only happen from the procedure editor?

In GUI, it happens with the procedure editor as well as without. But with the editor the message is displayed as an error alert-box, without the editor it is displayed in the message area with a pause.

I am not sure about ChUI, didn't test there.

#67 - 11/10/2016 12:50 AM - Vadim Gindin

The implemented case is the following. For CHUI WARNING is not shown. For GUI the WARNING is shown if "waited" menu-item belongs to a menu, that is already attached to a current-window, but there is no enabled frames. It is still actual.. I don't understand what Hynek means talking that "Frame states have no effect.."

It would be interesting to test the following cases:

1. What if MENU is attached to some other window, not current?
2. What if "waited" menu-item belongs to popup menu (attached to a widget from the current/not current window, or also not attached)
3. What if "waited" widget is not menu element at all.

#68 - 11/10/2016 12:58 AM - Vadim Gindin

I've committed missed history entries.

#69 - 11/10/2016 03:54 AM - Hynek Cihlar

Vadim Gindin wrote:

The implemented case is the following. For CHUI WARNING is not shown. For GUI the WARNING is shown if "waited" menu-item belongs to a menu, that is already attached to a current-window, but there is no enabled frames. It is still actual.. I don't understand what Hynek means talking that "Frame states have no effect.."

Vadim, please try the examples in notes 56 and 59. They are in contrary with the algorithm you describe.

Below another example, menu item belongs to a menu attached to a window, and no enabled frames. In native 4GL **no** warning/error is displayed, which is in contrary to your implementation.

```
define menu mbar menubar
  MENU-ITEM quitit label "E&xit".

assign current-window:menubar = menu mbar:handle.

DEF VAR i AS CHAR.
DEFINE FRAME f i.
VIEW FRAME f.

wait-for choose of menu-item quitit.
```

Clearly if you take a look at the examples I've given you can rule out the frames from the equation. The warning I believe relates to the fact that the waitable widget (menu-item in this case) is in the correct wait state - attached to a menu, menu attached to a window and the window **realized**.

It would be interesting to test the following cases:

1. What if MENU is attached to some other window, not current?
2. What if "waited" menu-item belongs to popup menu (attached to a widget from the current/not current window, or also not attached)
3. What if "waited" widget is not menu element at all.

Yes I think these are perfectly valid questions and potential cases that should be investigated.

#70 - 11/10/2016 07:45 AM - Greg Shah

Hynek: Does the example in note 69 generate the warning?

Vadim: Is it true that menus in the 4GL don't need to be ENABLEd before they can accept user input?

#71 - 11/10/2016 07:55 AM - Vadim Gindin

Greg Shah wrote:

Hynek: Does the example in note 69 generate the warning?

Vadim: Is it true that menus in the 4GL don't need to be ENABLEd before they can accept user input?

Menus are "enabled" by default. To make some MENU-ITEM disabled we can add DISABLED to definition. Why are you asking?

#72 - 11/10/2016 08:10 AM - Greg Shah

Menus are "enabled" by default.

I'm asking so I understand the testcases posted here. Since menus are enabled by default, I assume you can have a window with menus and without any frames. And in that case there is no warning and the user can use the menus. Correct?

#73 - 11/10/2016 08:21 AM - Vadim Gindin

Greg Shah wrote:

Menus are "enabled" by default.

I'm asking so I understand the testcases posted here. Since menus are enabled by default, I assume you can have a window with menus and without any frames. And in that case there is no warning and the user can use the menus. Correct?

Yes, we can have a window with menus and without any frames. But see the test from the note 43. Comment string MESSAGE "SOME TEXT" and run. You'll see the WARNING

#74 - 11/10/2016 08:35 AM - Greg Shah

Yes, we can have a window with menus and without any frames. But see the test from the note 43. Comment string MESSAGE "SOME TEXT" and run. You'll see the WARNING

What about the example from note 69? Does it generate the warning?

Please do test the suggested scenarios from note 67 and post your results here.

#75 - 11/10/2016 08:46 AM - Vadim Gindin

Greg Shah wrote:

Yes, we can have a window with menus and without any frames. But see the test from the note 43. Comment string MESSAGE "SOME TEXT" and run. You'll see the WARNING

What about the example from note 69? Does it generate the warning?

No. And If we comment line assign current-window:menubar = menu mbar:handle PROGRESS will show the WARNING.

Please do test the suggested scenarios from note 67 and post your results here.

OK.

#76 - 11/10/2016 08:47 AM - Hynek Cihlar

Greg Shah wrote:

Hynek: Does the example in note 69 generate the warning?

For the example in note 69 no warning is shown in native 4GL, in P2J the subjected warning is shown.

#77 - 11/10/2016 08:53 AM - Hynek Cihlar

Vadim Gindin wrote:

Greg Shah wrote:

Menus are "enabled" by default.

I'm asking so I understand the testcases posted here. Since menus are enabled by default, I assume you can have a window with menus and without any frames. And in that case there is no warning and the user can use the menus. Correct?

Yes, we can have a window with menus and without any frames. But see the test from the note 43. Comment string MESSAGE "SOME TEXT" and run. You'll see the WARNING

When you comment out the MESSAGE statement the window will not get displayed, this will make the menu item "non-waitable" and hence the warning. Note that enabling a frame makes its window visible, too. This could make you believe the frame is the cause, but it is not.

#78 - 11/10/2016 08:57 AM - Vadim Gindin

Hynek Cihlar wrote:

When you comment out the MESSAGE statement the window will not get displayed, this will make the menu item "non-waitable" and hence the warning.

Could you clarify why it makes menu-item "non-waitable"?

#79 - 11/10/2016 09:02 AM - Greg Shah

And If we comment line assign current-window:menubar = menu mbar:handle PROGRESS will show the WARNING.

This reason is simple: the WAIT-FOR references a menu-item widget that cannot be accessed because its menubar is not attached to a window. Does P2J show the warning in this case?

#80 - 11/10/2016 09:04 AM - Hynek Cihlar

Vadim Gindin wrote:

Hynek Cihlar wrote:

When you comment out the MESSAGE statement the window will not get displayed, this will make the menu item "non-waitable" and hence the warning.

Could you clarify why it makes menu-item "non-waitable"?

Non-waitable in this case means that when you reference the widget in WAIT-FOR you will get an error message from 4GL runtime.

#81 - 11/10/2016 09:05 AM - Greg Shah

Vadim Gindin wrote:

Hynek Cihlar wrote:

When you comment out the MESSAGE statement the window will not get displayed, this will make the menu item "non-waitable" and hence the warning.

Could you clarify why it makes menu-item "non-waitable"?

I think he is saying that without the MESSAGE statement (or a DISPLAY/VIEW...) the window will not be realized. An unrealized window cannot receive input events from a user. Because of this the WAIT-FOR is invalid.

Hynek: Am I understanding your point correctly?

#82 - 11/10/2016 09:14 AM - Hynek Cihlar

Greg Shah wrote:

Vadim Gindin wrote:

Hynek Cihlar wrote:

When you comment out the MESSAGE statement the window will not get displayed, this will make the menu item "non-waitable" and hence the warning.

Could you clarify why it makes menu-item "non-waitable"?

I think he is saying that without the MESSAGE statement (or a DISPLAY/VIEW...) the window will not be realized. An unrealized window cannot receive input events from a user. Because of this the WAIT-FOR is invalid.

Hynek: Am I understanding your point correctly?

Exactly, this is the point.

#83 - 11/10/2016 09:17 AM - Vadim Gindin

Hynek Cihlar wrote:

Greg Shah wrote:

Vadim Gindin wrote:

Hynek Cihlar wrote:

When you comment out the MESSAGE statement the window will not get displayed, this will make the menu item "non-waitable" and hence the warning.

Could you clarify why it makes menu-item "non-waitable"?

I think he is saying that without the MESSAGE statement (or a DISPLAY/VIEW...) the window will not be realized. An unrealized window cannot receive input events from a user. Because of this the WAIT-FOR is invalid.

Hynek: Am I understanding your point correctly?

Exactly, this is the point.

In other words, window (got by wait-for widget) must be realized at the moment of wait-for call. Right?

#84 - 11/10/2016 09:24 AM - Vadim Gindin

Vadim Gindin wrote:

Hynek Cihlar wrote:

Greg Shah wrote:

Vadim Gindin wrote:

Hynek Cihlar wrote:

When you comment out the MESSAGE statement the window will not get displayed, this will make the menu item "non-waitable" and hence the warning.

Could you clarify why it makes menu-item "non-waitable"?

I think he is saying that without the MESSAGE statement (or a DISPLAY/VIEW...) the window will not be realized. An unrealized window cannot receive input events from a user. Because of this the WAIT-FOR is invalid.

Hynek: Am I understanding your point correctly?

Exactly, this is the point.

In other words, window (got by wait-for widget) must be realized at the moment of wait-for call. Right?

Not like this. Like this. We have a widget from WAIT-FOR. We're trying to get a window from it. If result window is null or it is not realized => show WARNING.

#85 - 11/10/2016 09:26 AM - Hynek Cihlar

Vadim Gindin wrote:

In other words, window (got by wait-for widget) must be realized at the moment of wait-for call. Right?

Yes.

#86 - 11/10/2016 09:42 AM - Vadim Gindin

Greg, Hynek.

I agree with that.

I've checked noted points superficially:

What if MENU is attached to some other window, not current?

What if "waited" menu-item belongs to popup menu (attached to a widget from the current/not current window, or also not attached)

What if "waited" widget is not menu element at all.

In all tests proposed criteria is suitable. Menus are not related to this WARNING at all. Important only a window taken from "waitable" widget. If it null or not realized => show warning.

By the way, As I remember frames can be nested..

#87 - 11/14/2016 06:41 AM - Vadim Gindin

1. There is one problem: frame is already displayed at the moment of wait-for processing, but if we're going to show the warning we must have clean window. See the script from the note 69 for example.
2. wait-for can contain multiple widgets: needs more testing.

#88 - 11/15/2016 05:10 AM - Vadim Gindin

Vadim Gindin wrote:

1. There is one problem: frame is already displayed at the moment of wait-for processing, but if we're going to show the warning we must have

clean window. See the script from the note 69 for example.
2. wait-for can contain multiple widgets: needs more testing.

if wait-for contains several widgets - show the message only if all of them are "incorrect".

Is WAIT-FOR necessary statement in PROGRESS? Can we guarantee, that it always exists?

#89 - 11/15/2016 11:44 AM - Greg Shah

There is one problem: frame is already displayed at the moment of wait-for processing, but if we're going to show the warning we must have clean window.

What do you mean by a "clean window"?

Why is this a problem?

if wait-for contains several widgets - show the message only if all of them are "incorrect".

This is the result of your testing in the 4GL?

What do you mean by "incorrect"?

Is WAIT-FOR necessary statement in PROGRESS?

It is necessary for normal widgets to react to user input. For widgets like FILL-IN this means they can be edited. For widgets like a button it means they can receive and respond to events (like a mouse click).

Is this true for menus too?

Can we guarantee, that it always exists?

No, it doesn't have to exist. But if it is missing, then the widgets can only be seen, but the user cannot interact with them.

Why do you ask?

#90 - 11/15/2016 12:59 PM - Vadim Gindin

Greg Shah wrote:

There is one problem: frame is already displayed at the moment of wait-for processing, but if we're going to show the warning we must have clean window.

What do you mean by a "clean window"?

When the WARNING is shown the window is empty (clean) in spite of menubar is attached or frame is defined and viewed earlier (see below).

Why is this a problem?

Have a look at the test from the not 69. In spite of VIEW frame f. exists before WAIT-FOR we must not show the frame if quitit belongs to menubar which is not attached to a window. To see that just comment ASSIGN statement.

if wait-for contains several widgets - show the message only if all of them are "incorrect".

This is the result of your testing in the 4GL?

Yes.

What do you mean by "incorrect"?

I mean the definition noted earlier: widget is attached to a hidden window or not attached to window.

Is WAIT-FOR necessary statement in PROGRESS?

It is necessary for normal widgets to react to user input. For widgets like FILL-IN this means they can be edited. For widgets like a button it means they can receive and respond to events (like a mouse click).
Is this true for menus too?

You probably ask about MENU widget. Yes, it can respond to events. For example it can respond to key event to process possible item accelerator and find appropriate item. Mouse click event is responded for popup menus to close menu if the click was made somewhere outside of it.

Can we guarantee, that it always exists?

No, it doesn't have to exist. But if it is missing, then the widgets can only be seen, but the user cannot interact with them.

Why do you ask?

At this moment when we process VIEW frame f or ENABLE frame f we show the window if it hidden (if I'm right). But now we must delay somehow viewing the frame or enabling the frame until we check WAIT-FOR widgets whether they are "correct".

#91 - 11/15/2016 01:05 PM - Vadim Gindin

I forgot to add.. I should start with that: when I'm running the test from the note 69 (ASSIGN is commented) in P2J: I see the frame and the WARING simultaneously. That is the problem. It shows the frame and after that it processes WAIT-FOR and shows the WARNING.

#92 - 11/15/2016 02:01 PM - Greg Shah

At this moment when we process VIEW frame f or ENABLE frame f we show the window if it hidden (if I'm right). But now we must delay somehow viewing the frame or enabling the frame until we check WAIT-FOR widgets whether they are "correct".

This doesn't make sense to me. In the note 69 case, there is no warning. I thought we already determined that it is the unrealized window that causes the warning. If you have a VIEW FRAME f or an ENABLE FRAME f, then the window will be realized. If those are done before the WAIT-FOR, then you won't get a warning. What am I missing?

#93 - 11/15/2016 02:30 PM - Vadim Gindin

Greg Shah wrote:

At this moment when we process VIEW frame f or ENABLE frame f we show the window if it hidden (if I'm right). But now we must delay somehow viewing the frame or enabling the frame until we check WAIT-FOR widgets whether they are "correct".

This doesn't make sense to me. In the note 69 case, there is no warning. I thought we already determined that it is the unrealized window that causes the warning. If you have a VIEW FRAME f or an ENABLE FRAME f, then the window will be realized. If those are done before the WAIT-FOR, then you won't get a warning. What am I missing?

Did you comment the line: ASSIGN current-window:menubar?

#94 - 11/15/2016 03:39 PM - Greg Shah

Did you comment the line: ASSIGN current-window:menubar?

I haven't had time to try this myself, I am busy with other deadlines.

If there is no menubar, then the menu widget is invalid BUT there is no relationship between the menu widget and the window. Doesn't a VIEW FRAME f or an ENABLE FRAME f in that case cause the frame f to appear?

In other words, if you do this:

```
define menu mbar menubar
  MENU-ITEM quitit label "E&xit".

/* assign current-window:menubar = menu mbar:handle. */

DEF VAR i AS CHAR.
DEFINE FRAME f i.
VIEW FRAME f.

PAUSE.

wait-for choose of menu-item quitit.
```

In this case, doesn't the frame appear before the warning?

#95 - 11/15/2016 10:51 PM - Vadim Gindin

Greg Shah wrote:

Did you comment the line: ASSIGN current-window:menubar?

I haven't had time to try this myself, I am busy with other deadlines.

If there is no menubar, then the menu widget is invalid BUT there is no relationship between the menu widget and the window. Doesn't a VIEW FRAME f or an ENABLE FRAME f in that case cause the frame f to appear?

In other words, if you do this:

[...]

In this case, doesn't the frame appear before the warning?

Yes, the frame appears and when I press some key it disappears and the WARNING is shown.. When I press some key again, the frame appears again. So we should probably hide all frames before the WARNING.

#96 - 11/16/2016 04:07 AM - Hynek Cihlar

Vadim Gindin wrote:

Yes, the frame appears and when I press some key it disappears and the WARNING is shown.. When I press some key again, the frame appears again. So we should probably hide all frames before the WARNING.

I think the reappearance is caused by the program being restarted. It seems that the error manifesting as "Run-time error in WAIT-FOR..." aborts the current wait-for event loop and apparently in some cases abends the whole program.

```
MESSAGE "Test started".

define menu mbar menubar
  MENU-ITEM quitit label "Exit".

ASSIGN current-window:menubar = menu mbar:handle.

define menu bogus-mbar menubar
  MENU-ITEM bogus label "Bogus".

PROCEDURE foo:
  MESSAGE "foo start".
  wait-for choose of menu-item bogus IN MENU bogus-mbar.
  MESSAGE "foo finished".
END.

ON 'x':U ANYWHERE
DO:
  RUN foo.
END.

WAIT-FOR CHOOSE OF MENU-ITEM quitit.
MESSAGE "Test finished".
```

The following modification of the above only aborts the inner wait-for event loop.

```
MESSAGE "Test started".

define menu mbar menubar
  MENU-ITEM quitit label "Exit".

assign current-window:menubar = menu mbar:handle.

DEF VAR bogus AS HANDLE.
CREATE BUTTON bogus.

PROCEDURE foo:
  MESSAGE "foo start".
  wait-for choose of bogus.
  MESSAGE "foo finished".
END.

ON 'x':U ANYWHERE
DO:
  RUN foo.
END.

WAIT-FOR CHOOSE OF MENU-ITEM quitit.
MESSAGE "Test finished".
```

In both the examples press 'x' and then choose the 'Exit' menu item (if available).

#97 - 11/16/2016 05:04 AM - Vadim Gindin

What conclusion do you think it means?

#98 - 11/16/2016 05:11 AM - Hynek Cihlar

Vadim Gindin wrote:

What conclusion do you think it means?

It shows that the WAIT-FOR abort behavior may depend on other conditions, like the widget type. I think it will be required to also test the behavior of other widget types.

#99 - 11/16/2016 05:14 AM - Vadim Gindin

Hynek Cihlar wrote:

Vadim Gindin wrote:

What conclusion do you think it means?

It shows that the WAIT-FOR abort behavior may depend on other conditions, like the widget type. I think it will be required to also test the behavior of other widget types.

Makes it sense manually hide/show frames or as you've noted aborting is more appropriate?

#100 - 11/16/2016 05:19 AM - Hynek Cihlar

Vadim Gindin wrote:

Hynek Cihlar wrote:

Vadim Gindin wrote:

What conclusion do you think it means?

It shows that the WAIT-FOR abort behavior may depend on other conditions, like the widget type. I think it will be required to also test the behavior of other widget types.

Makes it sense manually hide/show frames or as you've noted aborting is more appropriate?

The frame's hide/show is a consequence of the program restart. So I don't think it is correct to manually hide or show frames as a result of the error condition. The correct approach seems to be to replicate the error behavior.

#101 - 11/16/2016 07:50 AM - Greg Shah

To determine the condition that is raised, you can wrap the WAIT-FOR like this:

```
do on <condition> undo, leave:
  wait-for ...
  message "after wait-for in do".
end.
message "after do".
```

Please check ERROR, ENDKEY, STOP and QUIT conditions.

The frame's hide/show is a consequence of the program restart.

If the frame is missing at the time the warning is displayed, then it is likely that the 4GL does some kind of clear all before the warning and then raises the condition.

#102 - 11/16/2016 07:59 AM - Hynek Cihlar

Greg Shah wrote:

The frame's hide/show is a consequence of the program restart.

If the frame is missing at the time the warning is displayed, then it is likely that the 4GL does some kind of clear all before the warning and then raises the condition.

Yes, this would make better sense.

#103 - 11/16/2016 09:49 AM - Vadim Gindin

Hynek Cihlar wrote:

Greg Shah wrote:

The frame's hide/show is a consequence of the program restart.

If the frame is missing at the time the warning is displayed, then it is likely that the 4GL does some kind of clear all before the warning and then raises the condition.

Yes, this would make better sense.

I've checked ENDKEY, ERROR, STOP and QUIT events. "after wait-for in do" wasn't printed in all cases. "after do" was printed only for STOP event. I.e. wait-for raises STOP event.

By the way, I've found that there is another one error right before "Runtime error ...": None of the widgets used in the WAIT-FOR statement are in a state (such as SENSITIVE) such that specified event can occur. WAIT-FOR terminated. (4123)

#104 - 11/16/2016 12:23 PM - Greg Shah

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

#105 - 11/16/2016 02:37 PM - Vadim Gindin

The problem is when we throw StopConditionException - the program is restarted and throw it again and so on infinitely, but we need at the second time pass successful.

I've committed my current changes. Revision is 11138. The check was changed and added method windowSafe() that returns parent window or null. I've added **TODO** about necessity to throw STOP condition.

#106 - 04/25/2018 05:15 AM - Greg Shah

- Related to Bug #3553: ChUI menus don't work added

Files

Screenshot at 2016-11-09 14_35_32.png	4.56 KB	11/09/2016	Hynek Cihlar
WARNING.png	3.64 KB	11/09/2016	Vadim Gindin