User Interface - Feature #3269

implement optional enhanced mode for ABL GUI which supports dynamic layout

03/28/2017 10:41 AM - Greg Shah

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

Description

Related issues:

Related to User Interface - Feature #3261: enhanced browse that can optionall...	**Closed**
Related to User Interface - Feature #3875: finish dynamic layout implementation	**New**

History

#1 - 03/28/2017 10:44 AM - Greg Shah

The past week we have been discussing (with a customer) various possible improvements to the ABL. These improvements are NOT meant to be compatible with the Progress implementation. Instead, we are proposing an optional mode (selected at runtime using configuration in the directory) in which we deliver an alternate widget implementation that provides features missing in the ABL.

For browse, see #3261.

One feature the customer desperately wants is to allow the browse to expand to fill space available in a window/frame when that frame is made larger. Optimally, it could be in response to the user resizing the window.

One idea is to make the browse respond well to this resizing/more space available scenario. We would let the rest of the widgets in a frame layout normally, but have the browse take up the extra space. The columns would expand to fit the data in them. The overall browse dimensions would be larger. This would be less work that making all widgets aware of dynamic sizing and might be a quick way to solve the most critical issue.

- What approaches could we take to make this work?
- How much effort would it take?

#2 - 03/28/2017 10:45 AM - Greg Shah

From Constantin:

> I've been thinking about this. I wonder what limits them to have this done on 4GL side? Do you have a screen/scenario for this?
>
> The idea would be to 'anchor' the widgets on the bottom/right browser's border so that their locations will be relative to the frame border (and not fixed) and after that just adjust the browser's dimension to fill the entire space. This approach could be done in 4GL and via FWD language extension, too.
>
> In FWD, the complex part will be to determine what is the layout from which 'anchored' coordinates can be computed. Also, is this related to their window maximize re-layout behavior?
>
> To give an estimate, I'm not really sure. We will need to 1. determine the layout rules for this mode. and 2. implement it. Each has its own

complexity...

#3 - 03/28/2017 10:46 AM - Greg Shah

From Hynek:

> as Constantin has noted, it would help to know the actual UI scenarios (i.e. UI screens and the intentions for them). Depending on the scenario anchoring the browse could influence other widgets on the same frame, for example when there were other widgets to the right or below the browse.
>
> In the simple cases, could the 4GL code could listen on the frame/window resize events and just set the browse size accordingly? If so this would be a matter of making sure the browse itself layouts properly after its dimensions are changed. I think this wouldn't be complicated to do if it already doesn't relayout.

#4 - 03/28/2017 11:06 AM - Greg Shah

The more complete way to do this might involve the following:

1. Calculate (in advance) the layout that would happen for each frame at runtime. This is not foolproof because some dynamic settings can only be known at runtime. For the cases where we can do this, we save off this knowledge as additional frame configuration (in the frame def). The number of columns, relative width of the columns, whether the screen is oriented largely vertical/horizontal/square... all these things might be calculated. We can then develop heuristics for how to automatically layout and relayout these frames when there is more space.

2. Provide an ABL syntax enhancement to allow the programmer to explicitly specify:

- a "high level" layout strategy from some list of predefined strategies
- specific values that we would otherwise try to calculate (in item 1 above)

The syntax would convert to frame definition settings. Then we would use this configuration to implement a layout strategy at runtime.

3. Provide strong support for handling media queries and changing layout behavior appropriately for each type of media query. For example, a two column layout that has roughly equal sizes can be displayed in a single column layout for narrow screens like phones.

#5 - 07/11/2018 06:10 AM - Hynek Cihlar

- Assignee set to Hynek Cihlar

- Status changed from New to WIP

#6 - 07/11/2018 06:54 AM - Greg Shah

The initial pass at this is focused on supporting the dynamic resizing/layout of the enhanced browse (#3261).

- We need to be able to determine the optimal size from the browse. This would be based on the fonts and columns, we need to calculate this. It should be a supplemental set of values that are different from the original hard coded 4GL dimensions. If everything fits in the original 4GL dimensions, there is no need to force the size to be larger. But if more space could be used, then we should use it.
- Once the optimal browse size is known, then we can choose the best window size that can fit both the desktop and the browse (or as much of the browse as will fit). We also must consider the relative layout of the other widgets in relation to the browse, so that there is room for the entire screen.
- If the user resizes the window, then we need to react to that and redo the layout, allowing the browse to resize to fit best in the new window size without losing the other widgets.

#7 - 07/11/2018 07:27 AM - Hynek Cihlar

Greg Shah wrote:

> The initial pass at this is focused on supporting the dynamic resizing/layout of the enhanced browse (#3261).

In the initial pass can we simplify this only to a single browse in a tree of frames?

> - We need to be able to determine the optimal size from the browse. This would be based on the fonts and columns, we need to calculate this. It should be a supplemental set of values that are different from the original hard coded 4GL dimensions. If everything fits in the original 4GL dimensions, there is no need to force the size to be larger. But if more space could be used, then we should use it.

What is the column's optimal size? Is it the format definition, label size or max of both? By "original 4GL dimensions" I assume you mean the width and height defined at the browse definition.

> - Once the optimal browse size is known, then we can choose the best window size that can fit both the desktop and the browse (or as much of the browse as will fit). We also must consider the relative layout of the other widgets in relation to the browse, so that there is room for the entire screen.

Does it mean that the layout logic should also change the size of the parent top-level window? If so, should the window resize happen even when the window widget size is defined explicitly in 4GL?

> - If the user resizes the window, then we need to react to that and redo the layout, allowing the browse to resize to fit best in the new window size without losing the other widgets.

Can you please explain what you mean by "losing the other widgets"?

#8 - 07/11/2018 08:52 AM - Greg Shah

> In the initial pass can we simplify this only to a single browse in a tree of frames?

My worry with this is that the result on a screen with more than one browse might be poor. What is the savings for doing this?

> What is the column's optimal size? Is it the format definition, label size or max of both?

Probably the max of both, but perhaps Stanislav has some ideas here.

> By "original 4GL dimensions" I assume you mean the width and height defined at the browse definition.

Yes, any explicit dimensions set in the code OR if there are missing dimensions then the dimensions that the 4GL would have produced implicitly.

> Does it mean that the layout logic should also change the size of the parent top-level window? If so, should the window resize happen even when the window widget size is defined explicitly in 4GL?

Yes and yes. If we don't do this, then we will never see the difference from the browse being able to dynamically resize. I think almost every GUI window ever created in 4GL code has explicit sizing values.

We can always add 4GL extension syntax to disable auto-sizing at either the window or browse level.

> Can you please explain what you mean by "losing the other widgets"?

Placing them in a location that cannot be accessed by the user or which would move them off screen and require scrolling in the dynamic layout that was not needed in the static layout.

#9 - 07/11/2018 01:22 PM - Hynek Cihlar

Greg Shah wrote:

> In the initial pass can we simplify this only to a single browse in a tree of frames?

My worry with this is that the result on a screen with more than one browse might be poor. What is the savings for doing this?

If all the screens that will need the dynamic layout for the first pass only have one browser, we could save some effort. Multiple browse widgets will introduce new corner cases, especially when enclosed frames and virtual sizes are considered.

> Yes, any explicit dimensions set in the code OR if there are missing dimensions then the dimensions that the 4GL would have produced implicitly.
>
> > Does it mean that the layout logic should also change the size of the parent top-level window? If so, should the window resize happen even when the window widget size is defined explicitly in 4GL?
>
> Yes and yes. If we don't do this, then we will never see the difference from the browse being able to dynamically resize. I think almost every GUI window ever created in 4GL code has explicit sizing values.

If we forcefully change the window size we can break any app that intentionally layouts its windows. I've seen several enterprise desktop apps that open multiple window and layouts them automatically for the user into an optimal configuration on the screen.

IMHO it would be better to just let the user resize the window beyond the constraint values set in MAX-WIDTH-*/MAX-HEIGHT-*.

Also wouldn't the forceful window resize break a Web embedded layout?

#10 - 07/11/2018 01:55 PM - Hynek Cihlar

For the first pass I propose the following behavior.

The dynamic layout algorithm will keep the relative position of the widgets of the legacy layout. It will also keep the margins between widgets and the parent containers (i.e. frames). Any size delta of the parent window will be proportionally applied to all the contained widgets such that, size of the browse widgets will be changed proportionally and widgets will be moved to keep the relative positions and margins mentioned above. Proportional size change means that when multiple browse widgets overlap the same horizontal band, widths will be changed proportionally based on the initial layout. Likewise for the heights.

This should be simple to implement, with no 4GL extensions required. The downside is that there will be no media query or different layout strategies to change the widgets' relative positions. I.e. no support for small screens for example.

#11 - 07/11/2018 04:15 PM - Stanislav Lomany

> What is the column's optimal size? Is it the format definition, label size or max of both?

IMHO usually column data fits into the column because in most cases you're unable to change column's width. Have you ever seen cut cell data in an app? So my suggestion is to display as much columns and rows as possible. If all columns are displayed, we can proportionally enlarge them all.

#12 - 07/11/2018 04:24 PM - Greg Shah

> If all the screens that will need the dynamic layout for the first pass only have one browser, we could save some effort. Multiple browse widgets will introduce new corner cases, especially when enclosed frames and virtual sizes are considered.

I don't know if we can assume this. For now, plan to calculate the expansion for each browse on the window.

> If we forcefully change the window size we can break any app that intentionally layouts its windows. I've seen several enterprise desktop apps that open multiple window and layouts them automatically for the user into an optimal configuration on the screen.

True. That is why we would add some way to disable this feature (globally). It will only be enabled for a customer that chooses it. And we will add 4GL syntax to override this for specific windows.

All the 4GL GUI applications we've seen so far would benefit from this. None of them manage their windows in a careful layout. All of them are coded to a lower resolution screen and have hard coded values that are no longer good for newer high resolution setups. These small window setups combined with the fixed nature of the 4GL approach makes this feature important.

Please plan to force the window to a larger size.

> IMHO it would be better to just let the user resize the window beyond the constraint values set in MAX-WIDTH-*/MAX-HEIGHT-*.

This would require the user to take action every time they start a new window, which is annoying. Most of these applications have MANY windows so that is not a solution that will work.

> Also wouldn't the forceful window resize break a Web embedded layout?

Not really. The IFRAME defines the space in which the window can expand and the default setup for the IFRAME will be to take up most of the embedded mode space.

Also I think that the window will not always be resized to the maximum space available. This would only be the case if the optimal browse size cannot fit in the available space.

> The dynamic layout algorithm will keep the relative position of the widgets of the legacy layout. It will also keep the margins between widgets and the parent containers (i.e. frames).

Yes.

> Any size delta of the parent window will be proportionally applied to all the contained widgets such that, size of the browse widgets will be changed proportionally and widgets will be moved to keep the relative positions and margins mentioned above.

Yes. With the caveat that widgets to the left and/or above the browse would not be shifted. It is only widgets to the right and/or below the browse whose positions are affected.

> Proportional size change means that when multiple browse widgets overlap the same horizontal band, widths will be changed proportionally based on the initial layout. Likewise for the heights.

Yes.

What you are describing here is what I have been envisioning as well.

> This should be simple to implement, with no 4GL extensions required. The downside is that there will be no media query or different layout strategies to change the widgets' relative positions. I.e. no support for small screens for example.

I understand and agree. That is something I was not planning for this phase.

#13 - 07/12/2018 09:49 AM - Greg Shah

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

#14 - 07/18/2018 05:38 PM - Hynek Cihlar

Created task branch 3269a.

#15 - 07/31/2018 05:09 PM - Hynek Cihlar

3269a rebased against trunk revision 11275.

#16 - 08/02/2018 03:10 AM - Hynek Cihlar

3269a rebased against trunk revision 11276.

#17 - 08/02/2018 09:44 PM - Hynek Cihlar

- File Peek 2018-08-03 03-14.gif added

I'm attaching a demo demonstrating the current state of implementation. The attached animated gif shows the Hotel GUI app without any modifications to the legacy code.

The layout algorithm detects the expandable widgets (currently only browse is considered an expandable widget), finds the available space and resizes them accordingly. It moves the widgets around the expandable widgets to offset the resize of the expandable widgets and to make more room if available. The expandable widgets are resized proportionally based on their initial size. There is a configurable padding that is kept between widgets or between widgets and the top-level window's edge. Inner frames are supported as well as multiple browse widgets anywhere in the widget tree.

When browse is resized, the available space is distributed proportionally among all the (unlocked) columns. The proportion is calculated from the column's data width. There is a configurable minimum column size, defined as ratio of the column's data width. Minimum ratio of 1 will keep the column widths at minimum of their respective data widths, ratio of 0.5 will keep the minimum width at half of their respective widths.

Few notes to the attached animation:
(1) The resize breaks the UI somewhat. For example the Logout and Exit buttons stay put, but naturally they should move together with the right window edge. For this to work, the legacy code must be enriched with layout hints. For example in the form of an "anchor-*" frame-item options: anchor-top, anchor-left, anchor-bottom, anchor-right. So for the Logout and Exit buttons it would be enough to specify anchor-left and anchor-right.
(2) The window resize is painfully slow. This is also present in trunk. The problem is that we recalculate the layouts for each and every mouse event. If the full layout calculation was performed only for some mouse events, the resize would become more responsive.
(3) The window edges for mouse resize should be thicker. Currently in FWD they equal to the window border thickness and so in case of Window 10 theme where the border size is 1 pixel, it is difficult to aim with the mouse.

Some known issues:
(1) The optimal layout size is not yet implemented.
(2) Sometimes the frames are not relayouted when tabs are switched.
(3) Some screens cause artifacts in the tab control. I think this is caused by the layout padding logic. I will have to improve this or add more configuration options. For example, currently padding is added between enclosed frames, even when the parent frame only contains the child frame.

#18 - 08/03/2018 01:53 AM - Constantin Asofiei

Hynek Cihlar wrote:

> (2) The window resize is painfully slow. This is also present in trunk. The problem is that we recalculate the layouts for each and every mouse event. If the full layout calculation was performed only for some mouse events, the resize would become more responsive.

Do we want to redraw (and relayout) the resized window, after each pixel difference for windows' width/height? Can't we use a similar approach to how the 4GL does widget resize/move, and perform layout only once the resize has finished?

#19 - 08/03/2018 09:01 AM - Greg Shah

I think the results are fantastic. This is really, really good!

> (1) The resize breaks the UI somewhat. For example the Logout and Exit buttons stay put, but naturally they should move together with the right window edge. For this to work, the legacy code must be enriched with layout hints. For example in the form of an "anchor-*" frame-item options: anchor-top, anchor-left, anchor-bottom, anchor-right. So for the Logout and Exit buttons it would be enough to specify anchor-left and anchor-right.

I'm OK with this.

In the next phase of dynamic layout, I was planning frame-level and window-level layout strategies along with making all widgets layout aware. I guess this proposal fits pretty well into that model. What other widget-level options might be needed?

> > (2) The window resize is painfully slow. This is also present in trunk. The problem is that we recalculate the layouts for each and every mouse event. If the full layout calculation was performed only for some mouse events, the resize would become more responsive.
>
> Do we want to redraw (and relayout) the resized window, after each pixel difference for windows' width/height? Can't we use a similar approach to how the 4GL does widget resize/move, and perform layout only once the resize has finished?

I agree. Please go ahead with this.

#20 - 08/03/2018 09:23 AM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

> Hynek Cihlar wrote:
>
> > (2) The window resize is painfully slow. This is also present in trunk. The problem is that we recalculate the layouts for each and every mouse event. If the full layout calculation was performed only for some mouse events, the resize would become more responsive.
>
> Do we want to redraw (and relayout) the resized window, after each pixel difference for windows' width/height? Can't we use a similar approach to how the 4GL does widget resize/move, and perform layout only once the resize has finished?

Actually, this is not a 4LG 'feature', but a Windows solution for slow drawing applications. This is toggable from the Show window content while dragging in Performance Options of System Properties. But anyway, I also think that this is the solution for the moment: recalculate the layout only when the mouse is released, on all dragging events, only the guide-lines should be drawn (alternatively we can resize the window, but we freeze the content).

> > (3) The window edges for mouse resize should be thicker. Currently in FWD they equal to the window border thickness and so in case of Window 10 theme where the border size is 1 pixel, it is difficult to aim with the mouse.

I don't have a Win10 near me at the moment but I think they trick the user by switching mouse to RESIZE if the mouse is close enough to that pixel-width line.
Please use the Windows8 theme until we find a solution that best mimics the native feel.

#21 - 08/05/2018 12:57 PM - Hynek Cihlar

Greg Shah wrote:

> In the next phase of dynamic layout, I was planning frame-level and window-level layout strategies along with making all widgets layout aware. I guess this proposal fits pretty well into that model. What other widget-level options might be needed?

I don't see any options needed for this phase. But they may emerge when we start applying dynamic layout to real use cases.

With further development of the dynamic layout towards the responsive world, the additional options will be necessary. To specify layout strategy (simple grid, column layout, flexbox, etc.), to define the parameters for each of the layout strategy and last but not least options for media query.

> Do we want to redraw (and relayout) the resized window, after each pixel difference for windows' width/height? Can't we use a similar approach to how the 4GL does widget resize/move, and perform layout only once the resize has finished?

I agree. Please go ahead with this.

Ok.

#22 - 08/06/2018 12:05 PM - Hynek Cihlar

- File Peek 2018-08-06 17-58.mp4 added

Greg Shah wrote:

> Do we want to redraw (and relayout) the resized window, after each pixel difference for windows' width/height? Can't we use a similar approach to how the 4GL does widget resize/move, and perform layout only once the resize has finished?

I agree. Please go ahead with this.

It turned out to be significantly simpler to optimize the current resize with content relayout. See the attached video. I removed some unnecessary drawings caused by the window layout manager and only relayout with configurable time frames. The result is surprisingly smooth.

In Web GUI we already don't layout the window content when resizing. But I think this optimization technique would work well for Web GUI, too.

#23 - 08/06/2018 03:33 PM - Greg Shah

That is really impressive! Well done.

#24 - 08/13/2018 04:36 AM - Hynek Cihlar

3269a rebased against trunk revision 11278.

#25 - 08/13/2018 09:01 AM - Hynek Cihlar

Status report.

I am wrapping up regression testing. GUI passes all the regression tests. I found some errors during ChUI regression tests, these are cleared and I am now rerunning ChUI regression tests.

I am now finishing documentation and code comments. I will post review request once the comments are finished, I expect this later afternoon/evening local time.

#26 - 08/13/2018 06:38 PM - Hynek Cihlar

Please review 3269a revision 11303. The revision passed GUI regression tests, ChUI is yet in progress.

#27 - 08/14/2018 04:36 AM - Stanislav Lomany

> Please review 3269a revision 11303. The revision passed GUI regression tests, ChUI is yet in progress.

The browse part seems good.

#28 - 08/14/2018 08:00 AM - Hynek Cihlar

I should mention that the current state of the dynamic layout algorithm works by laying out widget containers individually. The consequence of this is that complex UIs with overlapping widgets where the logical relation of the widgets can't be determined from the widget structure will not layout properly. Unfortunately this is the case for one of the large customer's GUI app. These kind of UIs would require some assumptions of the widget structure or compile time hinting in order to give good layout results.

#29 - 08/14/2018 01:05 PM - Constantin Asofiei

Review for 3269a rev 11303:

- the -T- column needs to be removed from the header entries of Directory.java, DirectoryServer.java, NullDirectory.java
- OverlayWindow.java - you have an empty line before the history entry
- AbstractContainer.containerStateListeners is missing javadoc
- DynamicLayoutConfig is missing class javadoc

Otherwise, is the web client performance OK when resizing the window?

#30 - 08/14/2018 01:55 PM - Hynek Cihlar

Constantin Asofiei wrote:

> Review for 3269a rev 11303:
>
> - the -T- column needs to be removed from the header entries of Directory.java, DirectoryServer.java, NullDirectory.java
> - OverlayWindow.java - you have an empty line before the history entry
> - AbstractContainer.containerStateListeners is missing javadoc

- DynamicLayoutConfig is missing class javadoc

Thanks, will fix.

> Otherwise, is the web client performance OK when resizing the window?

Web GUI window resize was already handled by JS, the window resize request is not sent to the client until the mouse drag is finished. During mouse dragging a window overlay is shown. I noticed this only after I implemented the optimization for Swing. But since the optimization is implemented above the driver level it would be simple to integrate it in Web GUI, too.

#31 - 08/14/2018 05:44 PM - Hynek Cihlar

Hynek Cihlar wrote:

> Constantin Asofiei wrote:
>
> > Review for 3269a rev 11303:
> >
> > - the -T- column needs to be removed from the header entries of Directory.java, DirectoryServer.java, NullDirectory.java
> > - OverlayWindow.java - you have an empty line before the history entry
> > - AbstractContainer.containerStateListeners is missing javadoc
> > - DynamicLayoutConfig is missing class javadoc
>
> Thanks, will fix.

Fixed in 3269a revision 11306. I also checked in fixes for several layout edge cases. Please review. ChUI regression tests passed.

Except for the pending review I don't have any remaining tasks on this issue. The dynamic layout feature should be now confronted with the expected use cases.

#32 - 08/15/2018 11:43 AM - Eric Faulhaber

Hynek, Greg, Constantin, is there anything holding back merging 3269a to trunk?

Hynek, how do we enable the feature?

#33 - 08/15/2018 11:46 AM - Eric Faulhaber

Hynek Cihlar wrote:

> These kind of UIs would require some assumptions of the widget structure or compile time hinting in order to give good layout results.

Any thoughts on what this hinting would look like?

#34 - 08/15/2018 11:50 AM - Greg Shah

Eric Faulhaber wrote:

> Hynek Cihlar wrote:
>
> > These kind of UIs would require some assumptions of the widget structure or compile time hinting in order to give good layout results.
>
> Any thoughts on what this hinting would look like?

We've planned 4GL code changes to provide this.

#35 - 08/15/2018 11:51 AM - Greg Shah

> is there anything holding back merging 3269a to trunk?

I think it can be merged to trunk.

#36 - 08/15/2018 02:01 PM - Hynek Cihlar

Eric Faulhaber wrote:

> Hynek, Greg, Constantin, is there anything holding back merging 3269a to trunk?
>
> Hynek, how do we enable the feature?

Eric, there is are couple of new config values in the directory. I will post here a documentation describing this, together with some info how the feature works.

#37 - 08/15/2018 02:03 PM - Hynek Cihlar

Eric Faulhaber wrote:

> Hynek Cihlar wrote:
>
> > These kind of UIs would require some assumptions of the widget structure or compile time hinting in order to give good layout results.
>
> Any thoughts on what this hinting would look like?

Eric, see the ANCHOR-* mentions in #3269-17.

#38 - 08/15/2018 02:03 PM - Hynek Cihlar

Greg Shah wrote:

> is there anything holding back merging 3269a to trunk?
>
> I think it can be merged to trunk.

Greg, there is a pending review, see #3269-31.

#39 - 08/15/2018 02:48 PM - Constantin Asofiei

Hynek Cihlar wrote:

> Fixed in 3269a revision 11306. I also checked in fixes for several layout edge cases. Please review. ChUI regression tests passed.

I'm OK with the changes.

#40 - 08/15/2018 04:01 PM - Hynek Cihlar

3269a 11306 was merged to trunk as revision 11279. The task branch was archived.

#41 - 08/17/2018 11:55 AM - Eric Faulhaber

Hynek Cihlar wrote:

> Eric, there is are couple of new config values in the directory. I will post here a documentation describing this, together with some info how the feature works.

Hynek, please post today at least the minimum info needed to turn the feature on. I want to do some testing. Thanks.

#42 - 08/17/2018 12:28 PM - Hynek Cihlar

To enable the dynamic layout feature, it must be turned on in the directory. The feature is turned globally for all top-level windows (regular windows as well as modal windows).

Add the following to the directory under /server/default/runtime/default (or other supported nodes, see below) to enable the feature:

```
<node class="container" name="dynamic-layout">
  <node class="boolean" name="enabled">
    <node-attribute name="value" value="TRUE"/>
  </node>
</node>
```

The layout options are read from the following directory nodes, in the order given:
/server/<serverID>/runtime/<account_or_group>/dynamic-layout
/server/<serverID>/runtime/default/dynamic-layout
/server/<serverID>/runtime/dynamic-layout
/server/default/runtime/<account_or_group>/dynamic-layout
/server/default/runtime/default/dynamic-layout

Here are the available values:

Value Name	Type	Default	Meaning
enable	boolean	false	Enables dynamic layout feature. This value is ignored in ChUI, dynamic layout is only supported in GUI.

frame-padding-pixels	int	2	The padding in pixels between the individual sections on the screen. I.e. this is the space between browse and surrounding widgets.
window-padding-pixels	int	0	The padding in pixels between the widgets and window edges.
min-column-width-ratio	double	0.5	This is the minimal ratio for the browse column widths. The minimal column width is determined as the product of this value and the column data width.
optimal-size	boolean	false	Enables the optimal size feature. When true, the algorithm will calculate the optimal screen size and resize the top-level window accordingly. Note that this only works for non-modal windows.

When dynamic layout is on, the algorithm scans the window screen for widgets and divides it in sections. When it finds a resizable (aka dynamic) widget (currently only browse or frame containing browse) it will do its magic, otherwise it will keep the section intact. As a corollary, when no browse widget appears on the screen, no dynamic layout will happen on the screen.

When dynamic layout is on, the browse columns will always stretch to take the available width.

The dynamic layout is performed after the standard legacy layout is performed. The points where this happens are: frame is realized, window is resized.

#43 - 08/17/2018 02:43 PM - Eric Faulhaber

How do we get the dynamic layout feature to fill the available iframe space in embedded mode?

#44 - 08/17/2018 03:14 PM - Hynek Cihlar

Eric Faulhaber wrote:

> How do we get the dynamic layout feature to fill the available iframe space in embedded mode?

Is this available space still part of the legacy application?

#45 - 08/17/2018 03:46 PM - Constantin Asofiei

Eric Faulhaber wrote:

> How do we get the dynamic layout feature to fill the available iframe space in embedded mode?

The window would need to be maximized explicitly by the fwd-embedded-driver.p code. But we would need to consider non-browse modules (simple frames, for i.e. reports) - these cases would not fit for maximize.

Hynek Cihlar wrote:

> Is this available space still part of the legacy application?

Yes, the iframe is already being automatically resized to fill the entire area between the drawers and the header/footer.

#46 - 08/20/2018 10:27 AM - Hynek Cihlar

Constantin Asofiei wrote:

> Eric Faulhaber wrote:
>
> > How do we get the dynamic layout feature to fill the available iframe space in embedded mode?
>
> The window would need to be maximized explicitly by the fwd-embedded-driver.p code. But we would need to consider non-browse modules (simple frames, for i.e. reports) - these cases would not fit for maximize.
>
> Hynek Cihlar wrote:
>
> > Is this available space still part of the legacy application?

> Yes, the iframe is already being automatically resized to fill the entire area between the drawers and the header/footer.

Frankly I only tested the virtual web mode, I will look at the embedded case with Constantin's suggestions.

#48 - 10/05/2018 11:13 AM - Eric Faulhaber

- Status changed from WIP to Closed

- % Done changed from 0 to 100

It seems the core implementation is done. We can address defects and enhancements in separate tasks.

#49 - 01/11/2019 04:12 PM - Greg Shah

- Related to Feature #3875: finish dynamic layout implementation added

#50 - 06/01/2021 11:31 AM - Greg Shah

I've created Dynamic Layout based on #3269-42.

Hynek: Please review and edit as needed.

#51 - 06/01/2021 01:43 PM - Hynek Cihlar

Greg Shah wrote:

> I've created Dynamic Layout based on #3269-42.
>
> Hynek: Please review and edit as needed.

The wiki page looks good.

#52 - 06/01/2021 04:15 PM - Roger Borrello

Greg Shah wrote:

> I've created Dynamic Layout based on #3269-42.
>
> Hynek: Please review and edit as needed.

Should there be mention in the Wiki under which the 4GL can override the request for resizing? I cannot find the terminology, but if you want to

prevent the user from resizing windows, this enhancement won't override that method of prevention.

#53 - 06/01/2021 08:51 PM - Greg Shah

I added a note about the <window>:RESIZE attribute.

#54 - 06/02/2021 01:43 AM - Hynek Cihlar

Greg Shah wrote:

> I added a note about the <window>:RESIZE attribute.

It looks good.

#55 - 06/02/2021 07:51 AM - Greg Shah

Roger: Could you please capture some screen shots of example Windows before/after resize? I'd like 2 sequences (each with before/after):

- Hotel GUI Virtual Desktop Mode - Guests tab
- Hotel GUI Embedded Mode - Guests tab

Add these as examples into the wiki page, probably in the "How It Works" section. For the virtual desktop mode, you'll have to change the directory because dynamic layout is not enabled in that mode by default.

#56 - 06/02/2021 07:53 AM - Roger Borrello

Greg Shah wrote:

> Roger: Could you please capture some screen shots of example Windows before/after resize? I'd like 2 sequences (each with before/after):
>
> - Hotel GUI Virtual Desktop Mode - Guests tab
> - Hotel GUI Embedded Mode - Guests tab
>
> Add these as examples into the wiki page, probably in the "How It Works" section. For the virtual desktop mode, you'll have to change the directory because dynamic layout is not enabled in that mode by default.

Sure thing. I'll also add 1 video of the *action* for each, to see if that's effective.

#57 - 06/02/2021 11:49 AM - Roger Borrello

Take a look at Samples

It actually doesn't look correct, although I made the change noted.

#58 - 06/02/2021 11:56 AM - Greg Shah

It is definitely not correct behavior.

Also, you didn't do embedded mode. You did the Swing client. We don't need to show the Swing client. Please do embedded mode.

#59 - 06/02/2021 12:34 PM - Roger Borrello

Greg Shah wrote:

> It is definitely not correct behavior.
>
> Also, you didn't do embedded mode. You did the Swing client. We don't need to show the Swing client. Please do embedded mode.

Take another look. I don't have rights to remove hotel-gui_guests_unsized2.png, hotel-gui_guests_resized2.png and hotel-gui_guests_sizing2.png.mkv. Can you do so?

Also, what am I doing wrong with Virtual Desktop Mode? Should the directory update be in a different area than server/default?

#60 - 06/02/2021 12:37 PM - Roger Borrello

With respect to the sizing itself, the number of pixels that are used to determine if you are trying to grab diagonally must be very, very small. I could not actually get the cursor to grab the corner. I had to do 2 direct manipulations, one downward, and one sideways. Can that threshold be enlarged so that a corner could more easily be grabbed?

#61 - 06/02/2021 12:51 PM - Roger Borrello

Roger Borrello wrote:

> Also, what am I doing wrong with Virtual Desktop Mode? Should the directory update be in a different area than server/default?

I see... there was another entry in the hotel gui directory with it disabled. I wasn't aware there were options for desktop and embedded for the directory. It's not really mentioned above.

```
    <node class="container" name="server">
      <node class="container" name="default">
        <node class="container" name="runtime">
          <node class="container" name="default">
            <node class="container" name="dynamic-layout">
              <node class="container" name="desktop">
                <node class="boolean" name="enabled">
                  <node-attribute name="value" value="FALSE"/>
                </node>
                <node class="boolean" name="optimal-size">
                  <node-attribute name="value" value="FALSE"/>
                </node>
              </node>
              <node class="container" name="embedded">
```

```
            <node class="boolean" name="enabled">
              <node-attribute name="value" value="TRUE"/>
            </node>
            <node class="boolean" name="window-maximize">
              <node-attribute name="value" value="TRUE"/>
            </node>
            <node class="boolean" name="optimal-size">
              <node-attribute name="value" value="FALSE"/>
            </node>
          </node>
        </node>
```

Any comments on the optimal-size parameter? Should I leave it FALSE for the documentation?

#62 - 06/02/2021 01:00 PM - Greg Shah

> I don't have rights to remove hotel-gui_guests_unsized2.png, hotel-gui_guests_resized2.png and hotel-gui_guests_sizing2.png.mkv.

Done.

#63 - 06/02/2021 01:16 PM - Roger Borrello

Greg Shah wrote:

> > I don't have rights to remove hotel-gui_guests_unsized2.png, hotel-gui_guests_resized2.png and hotel-gui_guests_sizing2.png.mkv.
>
> Done.

Take another look at the updated documentation.

You can also remove hotel-gui_guests_resized1.png, hotel-gui_guests_unsized1.png, and hotel-gui_guests_sizing1.png.mkv

#64 - 06/02/2021 03:45 PM - Greg Shah

I like it. I also deleted the old files.

Files

Peek 2018-08-03 03-14.gif	2.63 MB	08/03/2018	Hynek Cihlar
Peek 2018-08-06 17-58.mp4	926 KB	08/06/2018	Hynek Cihlar