

## User Interface - Feature #2564

Feature # 2252 (Closed): implement GUI client support

### implement GUI BROWSE widget

04/30/2015 01:56 PM - Greg Shah

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Stanislav Lomany	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	GUI Support for a Complex ADM2 App	<b>vendor_id:</b>	GCD
<b>billable:</b>	No		
<b>Description</b>			
<b>Related issues:</b>			
Related to User Interface - Feature #2422: add features to BROWSE		<b>Closed</b>	<b>10/17/2014</b>
Related to User Interface - Bug #2799: browse widget doesn't execute validati...		<b>New</b>	
Related to User Interface - Bug #2796: Browse widget reports error in GUI mode		<b>New</b>	<b>10/27/2015</b>
Related to User Interface - Feature #2628: Non-fill-in column support in browse.		<b>WIP</b>	<b>07/30/2015</b>
Related to User Interface - Feature #2892: Implement browse scrolling with mo...		<b>Closed</b>	
Related to User Interface - Bug #2893: Mouse clicks on rows are not 100% accu...		<b>Closed</b>	
Related to User Interface - Feature #2894: Implement browse multiple selectio...		<b>Closed</b>	
Related to User Interface - Bug #2895: Implement separate COLUMN-LABEL annota...		<b>New</b>	<b>12/01/2015</b>
Related to User Interface - Bug #2596: Implement missing browse quirks.		<b>New</b>	<b>06/26/2015</b>
Related to User Interface - Feature #2959: Implement search by key in browse		<b>Closed</b>	<b>01/18/2016</b>
Related to User Interface - Feature #2936: Editing cell shrinking quirk		<b>New</b>	
Related to User Interface - Bug #2950: Wrong behavior of editing browse on ro...		<b>Closed</b>	<b>01/08/2016</b>
Related to User Interface - Feature #2630: Add ability to save new rows in br...		<b>New</b>	
Related to Database - Bug #2955: Update records participating in a scrolling ...		<b>WIP</b>	<b>01/13/2016</b>
Related to Database - Bug #2896: Close dynamically defined query when an unde...		<b>New</b>	<b>12/01/2015</b>
Related to User Interface - Feature #3038: implement browse attrs COLUMN-MOVA...		<b>Closed</b>	

## History

### #1 - 04/30/2015 02:05 PM - Greg Shah

The following options, attributes and methods were investigated in [#2422](#). They were determined to be GUI-specific. For this reason, the support for these must be added as part of this task.

#### Options

FIT-LAST-COLUMN  
NO-COLUMN-SCROLLING  
ROW-HEIGHT-CHARS  
COLUMN-BGCOLOR  
EXPANDABLE  
NO-SCROLLBAR-VERTICAL (should just be implemented as setting the SCROLLBAR-VERTICAL attribute false and letting that attribute's implementation be used)

#### Methods

MOVE-COLUMN

#### Attributes

ROW-HEIGHT-CHARS  
ALLOW-COLUMN-SEARCHING - this may be Windows-specific but our implementation must be there for any GUI  
SEPARATORS  
EXPANDABLE  
COLUMN-RESIZABLE  
LABEL-BGCOLOR

COLUMN-MOVABLE  
FIT-LAST-COLUMN  
MAX-CHARS  
SCROLLBAR-VERTICAL  
COLUMN-FGCOLOR  
NUM-ITERATIONS  
DOWN

**#2 - 05/01/2015 05:10 AM - Stanislav Lomany**

Addition to the list:

MIN-HEIGHT-CHARS  
SORT - I think that actually means implementation of drop-down cells

**#3 - 05/10/2015 02:51 AM - Stanislav Lomany**

Another addition: CREATE-ON-ADD.

**#4 - 08/28/2015 07:41 AM - Stanislav Lomany**

Attributes which affect drawing:

**Coloring:**

COLUMN-BGCOLOR attribute Column  
COLUMN-FGCOLOR attribute Column

BGCOLOR attribute Browse, Column, Cell  
FGCOLOR attribute Browse, Column, Cell

LABEL-BGCOLOR attribute Browse, Column  
LABEL-FGCOLOR attribute Browse, Column

SEPARATOR-FGCOLOR attribute Browse

**Sizing:**

MIN-COLUMN-WIDTH-CHARS attribute Browse  
MIN-COLUMN-WIDTH-PIXELS attribute Browse

WIDTH-CHARS attribute Browse, Column, Cell  
WIDTH-PIXELS attribute Browse, Column, Cell

HEIGHT-CHARS attribute Browse, Cell  
HEIGHT-PIXELS attribute Browse, Cell

ROW-HEIGHT-CHARS attribute Browse  
ROW-HEIGHT-PIXELS attribute Browse

FIT-LAST-COLUMN/EXPANDABLE attribute Browse  
NO-EMPTY-SPACE attribute Browse

**Fonts:**

FONT attribute Browse, Column, Cell  
LABEL-FONT attribute Browse, Column  
COLUMN-FONT attribute Column

**Decoration:**

MANUAL-HIGHLIGHT attribute Browse – not sure what it does, not applied to browse in the customer app  
ROW-MARKERS attribute Browse  
SENSITIVE attribute Browse  
SEPARATORS attribute Browse  
NO-BOX option  
NO-LABELS option

**#5 - 08/28/2015 07:46 AM - Stanislav Lomany**

Guys, you are mentioning "3D" mode for GUI widgets. How is it switched?

**#6 - 08/28/2015 08:20 AM - Greg Shah**

The \*-DCOLOR and \*-PFCOLOR attributes/options are supposed to only be for ChUI, so it makes some sense if they don't work.

**#7 - 08/28/2015 08:21 AM - Greg Shah**

Stanislav Lomany wrote:

Guys, you are mentioning "3D" mode for GUI widgets. How is it switched?

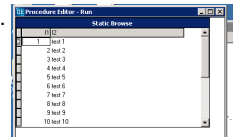
Are you thinking about the THREE-D attribute? It can be applied to the SESSION system handle, to a Window or to a frame/dialog. It has different behavior depending on where you set it.

**#8 - 08/31/2015 10:00 AM - Stanislav Lomany**

- File *browse.png* added

Guys, are there any existing components that I can re-use for drawing browse or that can simplify drawing process? I saw ScrollBarGuiImpl. Anything else?

It is always an integral number of rows in a browse, so main area is no "scrollable", I'll simply draw all visible rows from the top.



**#9 - 08/31/2015 10:02 AM - Stanislav Lomany**

For editing browse I'll re-use FillInGuiImpl component.

**#10 - 08/31/2015 10:37 AM - Greg Shah**

Hynek is working on major improvements to the scrolling support classes in task branch 2424c. It should be in testing today, but probably won't be finished until tomorrow. Don't spend any time on scrolling support until that code is in.

are there any existing components that I can re-use for drawing browse or that can simplify drawing process?

It seems like FillInGuiImpl would be useful for both editing as well as the drawing part. It is possible that the sizing of the columns is based on FILL-IN in the 4GL. We need to duplicate the column sizing exactly. The FillInGuiImpl already handles this for FILL-IN.

**#11 - 09/01/2015 10:59 AM - Greg Shah**

You should expect to use GuiFontResolver and GuiColorResolver.

**#12 - 09/01/2015 05:45 PM - Stanislav Lomany**

Some metrics. I'll update font\_details.odt accordingly.

Browse constraints:

	Width	Height
Minimum	1 char	2 browse rows
Maximum	650 chars	650 chars

Default browse size:

	Width	Height
Static	Should have place for all columns, can stretch parent frame.	-
Dynamic	30 chars	4 browse rows

Width of a column is COLUMN-FONT-WIDTH \* max(COLUMN-FORMAT-LENGTH, 2) for all data types. Checking if there are exceptions from this rule.

**#13 - 09/03/2015 12:20 PM - Stanislav Lomany**

Guys, browse column width depends on format width. Format width is sum of widths of its characters/placeholders. Character widths are different (except for fixed-width fonts). x 9 + - . , all have different widths. Difference slightly vary for different fonts. I think a way to solve it is to get widths of all format characters for all used fonts and all font sizes. I'm not sure how 4GL calculates width of "x" and "9" but it may be "average" font or digit width. Any better ideas to handle this?

**#14 - 09/03/2015 01:07 PM - Greg Shah**

Getting these text metrics should not be hard, since we already have a process and some tools for doing it.

To start, add the list of strings you need to get metrics for to frame\_generator.rules in the post-rules section. Look for uiStrings.add(":"). Use this same technique to add your list of strings.

1. Run conversion for all programs you need to support (they all must be run in the same batch). A file named ui\_strings.txt file will have your extra strings listed with all the other static text from the programs.
2. Copy the ui\_strings.txt file to windev01.
3. Copy p2j/tools/\* to windev01.
4. On windev01 run prowlin -p get-text-metrics.p.
5. Use ui\_strings.txt instead of lines-list.txt.
6. Let the program run to completion. It will create an output file named text-metrics.xml.
7. Copy text-metrics.xml to your P2J conversion system. Place it into the src/ directory of your project.
8. Compile/jar the converted code. The text-metrics.xml should be automatically placed in the resulting jar file.

At runtime, the font manager can now take advantage of the actual Windows sizing for these strings.

Constantin: did I leave anything out? Also: what other changes might be needed for him to leverage the results?

#### #15 - 09/06/2015 03:33 PM - Stanislav Lomany

Guys, consider the case in which field format is short, e.g. "x(1)". Browse column cannot be narrower than some font- and datatype-dependent width.

avg font width	min char width
2	5
6	11
7	14
10	16
13	26

There is no linear dependency of minimum column width with average or maximum font width. Do you have any ideas how to minimum column width can be calculated?

#### #16 - 09/06/2015 03:37 PM - Constantin Asofiei

Stanislav Lomany wrote:

Guys, consider the case in which field format is short, e.g. "x(1)". Browse column cannot be narrower than some font- and datatype-dependent width.

[...]

There is no linear dependency of minimum column width with average or maximum font width. Do you have any ideas how to minimum column width can be calculated?

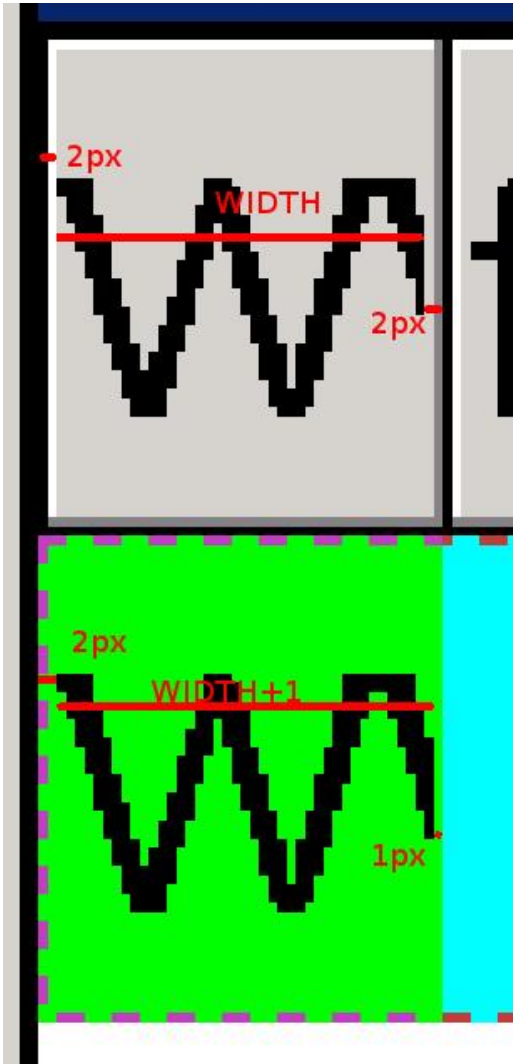
See the testcases/uast/fillin/CheckMetricsFillin.java and FillInGuiImpl.nativeWidth - for FILL-IN, the formula broken down to this:

```
// In GUI, the width has a deviation from the normal "multiply format width with the font's
// average character width" formula. After gathering fill-in metrics for various fonts and
// format lengths, it was determined that a special formula is used when all these are true:
// - the format length is between 1 and 12 characters
// - the font is not set explicitly via the FONT phrase
// - the widget normally expects a fixed font (like the case for a fill-in with a numeric
//   value)
// A java program used to prove this formula can be found in the testcases project,
// the uast/fillin/CheckMetricsFillin.java file.
int pixelWidth = !fixedFontFormat && !explicit && fmtWidth >= 1 && fmtWidth <= 12
    // handle font sizes 1 to 12
    ? Math.min(fmtWidth, 3) *
      (fixedFont && !explicit ? maxFontWidth
        : (avgFontWidth * 2 +
          (int) Math.floor(avgFontWidth / 2))) +
      Math.max(fmtWidth - 3, 0) * avgFontWidth

    // default case
    : fmtWidth * avgFontWidth;
```

It might be something similar for browse columns... the width for FILL-IN depends on the font type (fixed or not) and if it was explicitly set or not.

Browse column width metrics.



- WIDTH on the picture corresponds the value returned by WIDTH-PIXELS.
- WIDTH is maximum of format width and column label actual width.
- Format width is sum of its placeholders widths: X N A ! - average font width, 9 > Z \* - digit width (all digits seem to have equal widths), . , - + - actual symbol width.
- The last column has one pixel added to the right.

**#18 - 09/07/2015 12:06 PM - Greg Shah**

Good info.

What other research is needed before you are ready to start implementation? I do want you to understand things fully, up front. I just need to know the list of items still needing investigation.

**#19 - 09/07/2015 04:18 PM - Stanislav Lomany**

What other research is needed before you are ready to start implementation?

In terms of drawing there are some minor metrics remain.

Other features that need investigation are behavioral: horizontal scrolling, column resizing, column moving etc.

I think it is better to implement main functionality and then investigate new pieces of behavior as we get to them.

**#20 - 09/07/2015 08:11 PM - Greg Shah**

In terms of drawing there are some minor metrics remain.

Please get the metrics exactly right from the beginning. That has very high priority.

horizontal scrolling, column resizing, column moving etc.

Things like horizontal scrolling seem like they may be pretty important to the core functionality. It is OK to delay things like column resizing or column moving are not commonly used.

I'm OK with getting the main functionality done sooner, since it is critical for the demo. But I don't want to write a large amount of code that is wrong or which will need to be rewritten later. So things that affect the core functionality should be figured out now.



#21 - 09/09/2015 04:46 PM - Stanislav Lomany

- File browse1.png added

- File browse2.png added

Read-only browse metrics.

Browse header height is 20 px (with borders) (I haven't found a way to alternate title font). Header is not drawn if the title is not specified. You cannot set title together with no-box option.

Scrollbar width is 16 px.

Current row highlight: 3px/3px dotted line drawn clockwise from the left top corner.

If browse width is determined automatically then right border of the last column is not drawn

Separators are drawn as inner grid + right border. If there are no separators, 1px invisible grid between cells remain.

Row markers column's width depends on font width. I couldn't make 100% accurate formula, but it is very close to  $\text{floor}(\text{font\_width} * 1.8)$   
Row markers triangle is 4\*7px. Triangle offset left/top is  $\text{floor}((\text{width}/\text{height} - \text{triangle width}/\text{height} - 1) / 2)$

Column headers and rows height is font height + 3px.

ATTACHED PICTURE browse2.png IS INCORRECT - there should be 1px from top and 2px from bottom.

f1	f5
1	0
2	0
3	0
4	0
5	0
6	0

f1	f5
1	0
2	0
3	0
4	0
5	0
6	0
7	0

f1	f5
16	0
17	0
18	0
19	0
20	0

f1	f5
1	0
2	0
3	0
4	0
5	0

	III	f5
▶	1	0
	2	0
	3	0
	4	0
	5	0

2px + row height + 1 px

f1	f5
1	0
2	0
3	0
4	0
5	0

#22 - 09/09/2015 05:25 PM - Greg Shah

Good work.

Row markers column's width depends on font width. I couldn't make 100% accurate formula, but it is very close to  $\text{floor}(\text{font\_width} * 1.8)$

Constantin: Any ideas on how we determine this exactly?

Row markers triangle is 4\*7px. Triange offset left/top is  $\text{floor}((\text{width}/\text{height} - \text{triangle width}/\text{height} - 1) / 2)$

Please see [#2678](#). I expect that support to be checked in before Saturday. You would likely be able to reuse the same support as sb-arr-en-right.svg or at least the same approach.

**#23 - 09/09/2015 05:36 PM - Constantin Asofiei**

Row markers column's width depends on font width. I couldn't make 100% accurate formula, but it is very close to  $\text{floor}(\text{font\_width} * 1.8)$

Stanislav, which font are you referring to? DEFAULT-FONT/DEFAULT-FIXED-FONT or an explicit font set at the BROWSE? USE-3D setting in progress.ini usually adds 2 pixels to the width, so keep this in mind too.

You can try changing the DEFAULT-FONT/DEFAULT-FIXED-FONT in progress.ini and see how the browse looks. Try very small and very large font sizes :)

Also, you may want to check if is not something related to max-char-width for the font.

I guess is pretty difficult (if not impossible) to capture/interrogate metrics automatically for each browse sub-component, but if you haven't done already, please double-check the formulas you find with more than one font. The list of fonts we currently use is:

MS Sans Serif  
Courier New  
Segoe UI  
Microsoft Sans Serif  
System  
FixedSys  
Tahoma

**#24 - 09/09/2015 07:30 PM - Stanislav Lomany**

Stanislav, which font are you referring to? DEFAULT-FONT/DEFAULT-FIXED-FONT or an explicit font set at the BROWSE?

Explicit browse font.

USE-3D setting in progress.ini usually adds 2 pixels to the width, so keep this in mind too.

How do you use make progress use custom progress.ini? -iname option doesn't seem to work.

#25 - 09/10/2015 02:04 AM - Constantin Asofiei

Stanislav Lomany wrote:

Stanislav, which font are you referring to? DEFAULT-FONT/DEFAULT-FIXED-FONT or an explicit font set at the BROWSE?

Explicit browse font.

USE-3D setting in progress.ini usually adds 2 pixels to the width, so keep this in mind too.

How do you use make progress use custom progress.ini? -iname option doesn't seem to work.

The command is: `prowin32 -iname progress.ini -basekey INI -p your-program.p`

#26 - 09/10/2015 04:16 PM - Stanislav Lomany

- File *browse-edit1.png* added

I found no metrics differences in 3D and non-3D modes.  
DEFAULT-FONT/DEFAULT-FIXED-FONT doesn't affect browse header.

Behavior of in-browse fill-in matches behavior of a usual fill-in (not yet properly implemented in P2J).

	f1	f2	f3
10		test	true
20		test	true
30		test	true

	f1	f2	f3
10	test		true
20	test		true
30	test		true

	f1	f2
10	test	
20	test	
30	test	
40	test	
50	test	

#27 - 09/11/2015 07:50 AM - Greg Shah

usual fill-in (not yet properly implemented in P2J)

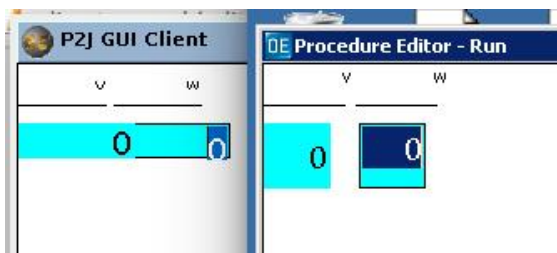
Please provide more details on your findings about what is not yet properly implemented.

Is there anything more to research for the main functionality? If not, how quickly can you have a working version of it?

#28 - 09/11/2015 09:38 AM - Stanislav Lomany

- File compare.png added

Please provide more details on your findings about what is not yet properly implemented.



Here is a simple testcase.

1. Are width and height fill-in differences expected?
2. You can see the highlighted area in the numeric field in 4GL to the left of 0. There are actually spaces and you can navigate thru them.

Is there anything more to research for the main functionality? If not, how quickly can you have a working version of it?

I'm done with research. I'll try to fit main functionality in a week.

### #29 - 09/15/2015 04:52 PM - Stanislav Lomany

Created task branch 2564a from P2J trunk revision 10936.

### #30 - 09/16/2015 06:47 PM - Stanislav Lomany

Guys, how can I set my fonts in P2J? I've added this to the directory:

```
<node class="container" name="server">
  <node class="container" name="default">
    <node class="container" name="font-table">
      <node class="string" name="font6">
        <node-attribute name="value" value="FixedSys, size=9"/>
      </node>
      <node class="string" name="font7">
        <node-attribute name="value" value="Segoe UI, size=14"/>
      </node>
      <node class="string" name="font8">
        <node-attribute name="value" value="Tahoma, size=24"/>
      </node>
    </node>
  </node>
</node>
```

These font specifications are read but eventually have no effect.

### #31 - 09/17/2015 02:55 AM - Constantin Asofiei

Stanislav Lomany wrote:

Guys, how can I set my fonts in P2J? I've added this to the directory:

[...]

These font specifications are read but eventually have no effect.

Note that bitmap fonts (like FixedSys), can't work in P2J yet. If you want to test if drawing/dimensions are OK, use Tahoma/Segoe UI/Courier New - there should be .ttf files in the simple/server/fonts folder.

### #32 - 09/18/2015 07:46 AM - Stanislav Lomany

Since it is a browse-related conversion issue, I decided to put it here.

The issue is about case sensitivity. This is the failing testcase:

```
DEFINE temp-table tt FIELD f1 as character.
```

```
DEFINE QUERY q FOR tt.  
DEFINE BROWSE BRWS QUERY q DISPLAY tt.f1 WITH 2 DOWN.  
DEFINE FRAME D-Dialog BRWS WITH SIZE 100 BY 20.  
  
ASSIGN tt.f1:BGCOLOR IN BROWSE brws = 10.
```

Change "brws" to "BRWS" in the last line and it will be converted.

For this issue was created the task branch 2564b from P2J trunk revision 10937.

### #33 - 09/23/2015 10:33 AM - Stanislav Lomany

- File *fillin-diff.png* added

- File *browse-cell-offset.ods* added

I forgot about one metrics. Vertical text positioning in a cell. Unfortunately P2J doesn't solve the similar problem for fill-in properly (see picture)



So I made some investigations (see the document) and as a result:

1. Dependency font size (not height) -> bottom offset is the most stable and linear.
2. Dependency formula is font-dependent, so for most accuracy it should have different formulas for all supported fonts.

### #34 - 09/23/2015 12:10 PM - Greg Shah

Do you have any idea if these vertical positioning rules are the same as for fill-in?

We already know of another case where we are handling the vertical position incorrectly, see [#2730](#).

### #35 - 09/23/2015 12:35 PM - Stanislav Lomany

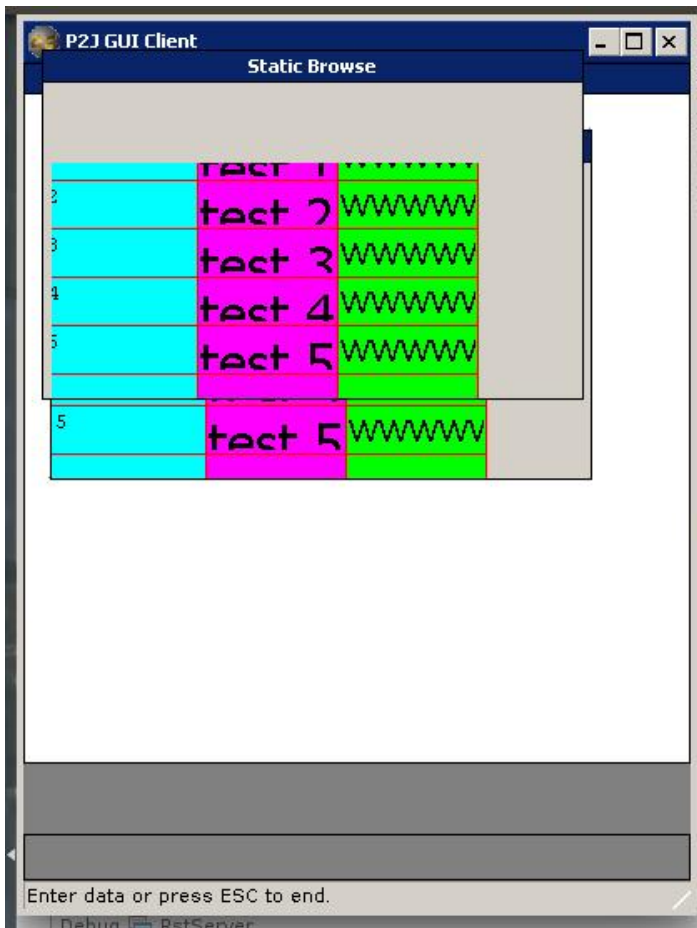
Do you have any idea if these vertical positioning rules are the same as for fill-in?

Brief investigations showed that offset values are different. Although general rules well may be the same.

### #36 - 09/27/2015 01:22 PM - Stanislav Lomany

- File *double-draw.png* added

Guys, there is an issue: when I call `draw()` and `requestSync()` on mouse/key event, browse offset is sometimes calculated from the top window instead of the parent frame. Do you have any ideas?



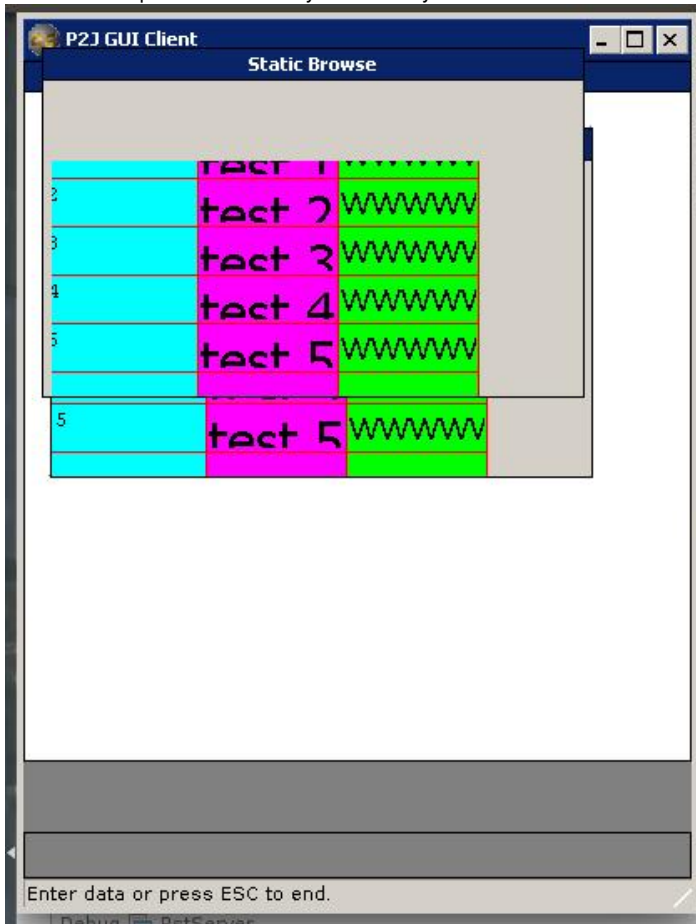
#37 - 09/27/2015 01:23 PM - Constantin Asotiei

Stanislav Lomany wrote:

Guys, there is an issue: when I call draw() and requestSync() on mouse/key event, browse offset is sometimes calculated from the top window



instead of the parent frame. Do you have any ideas?



requestSync needs to be overridden so that it calls repaint() - do not call draw() directly from the TC event processing loop.

- File w.png added

Reference text clipping:



#39 - 09/29/2015 10:11 AM - Stanislav Lomany

- File selection-color.png added

Guys, do we have a function that returns color of the specific pixel? I'm asking because dotted line that highlights the current row is colored according to the underlying pixels. This is a problem in the case this line crosses a cell text:



#40 - 09/29/2015 10:59 AM - Greg Shah

do we have a function that returns color of the specific pixel? I'm asking because dotted line that highlights the current row is colored according to the underlying pixels. This is a problem in the case this line crosses a cell text:

Interesting. No, we don't have such a function.

I believe that Progress wrote custom WIN32 code to implement the browse. I expect that when they implemented the row selection highlight, they found that a color that is close in shade to the text color needed better contrast where it overlaps the text. They probably implemented some standard compositing approach using WIN32 features when they drew a stroked rectangle for this selection highlight. Then they turn off that compositing after the rectangle is drawn.

I think we would want to allow the driver to handle this behavior. Otherwise, we would have to deliver too much pixel-level data to the browse implementation. This is a very low level thing and it isn't something for our higher level widget code (like browse).

I think the first step here is to try to identify the WIN32 compositing approach that is used.

At this time we only need this compositing feature for this stroked rectangle. That suggests that one approach is to make it a configurable option of the line stroke itself. Unless we know of other cases where we will need compositing control, I prefer to "hide" it in this specific feature. Implementing general purpose compositing is quite a lot of work and I don't want to take that on right now if it isn't needed. So I propose updating the LineStroke types to include a compositing configuration option.

Then the driver will be responsible for proper implementation of that feature.

In Java2D, I don't think we have direct access to the pixels. But we do have some standard Composite features that can be temporarily enabled. Hopefully we can match the WIN32 compositing using a standard Java approach.

In the web client, it is possible that we can try to leverage the compositing features of canvas. The problem is that there are (broadly) two different sets of compositing implementations which differ wildly in their visual results. In other words, Firefox and IE can be configured with the same compositing strategy but will yield different visual results for many of the w3c defined compositing approaches. I simply don't know if this will work.

On the other hand, we do already have full access to the pixels of the canvas itself. Thus, in the worst case we can manually composite. I'd like to avoid that, but we may not have a choice.

#### **#41 - 09/30/2015 08:42 AM - Stanislav Lomany**

Guys, I've subclassed ScrollContainer and added tried to scroll it with ScrollPaneGuiImpl. Container is scrolled ok, but the child widgets inside the container are not scrolled. Is that an expected behavior? Is that is supposed to be fixed?

#### **#42 - 09/30/2015 09:08 AM - Stanislav Lomany**

Greg, please consider the other approach for "inverse" coloring (it's not actually inverse but let's call it so). Of course I like the idea of being able to draw a rectagle with a single command with complexity of composing under the hood. But there is another, less elegant but, I think, more general approach: we do know the line colors for horisontal rectangle lines, so its no problem do draw them per cell. For vertical lines we can draw the text of the cell in using very small clip rectangles matching line strokes. Overhead is 10-20 text draw calls for both sides (or we can cache the bitmap).

#### **#43 - 09/30/2015 09:17 AM - Hynek Cihlar**

Stanislav Lomany wrote:

Guys, I've subclassed ScrollContainer and added tried to scroll it with ScrollPaneGuiImpl. Container is scrolled ok, but the child widgets inside the container are not scrolled. Is that an expected behavior? Is that is supposed to be fixed?

Is the change checked in?

What does it mean "container is scrolled ok", do you see the scroll bars move for example?

For vertical lines we can draw the text of the cell in using very small clip rectangles matching line strokes. Overhead is 10-20 text draw calls for both sides (or we can cache the bitmap).

Interesting idea. Actually, in your example image above it would only take 4 extra text drawing operations. I'm assuming that the original text was drawn first AND then the entire vertical selection highlight can be drawn AND then we would overdraw only the intersecting vertical selection highlight portions with the very tiny clip rectangles. In your example there are 4 different intersecting line segments, so that is the number of extra text draws needed.

We also would never need to overdraw more than a single character.

The problem here is that without being able to read the pixels in that vertical line, we can't detect the exact intersections of the line segments and the character. In Java we don't have access to the pixels. So we would have to overdraw the character for as many different line segments as that vertical stroked line has.

Another problem: Java hides the line stroking logic from us. In other words, in Java, how would we know how to set the clip rectangles to overdraw those segments?

On the positive side is that it avoids the potentially messy world of compositing, which is made harder by trying to be compatible using 2 different UI foundations (Swing and HTML5 Canvas).

On the negative side:

1. It encodes quite a bit of knowledge (about the pixels drawn for the stroked line) into the higher levels of our client. We are really trying to avoid that.
2. It is quite inefficient since the text drawing is very slow and we have to overdraw many times.

But I think that the limitations of the Java approach may make this idea less than easily accomplished.

Some thoughts:

1. Regardless of the approach, we need to be able to detect when this case will occur. Otherwise we wouldn't want to add any operations.
2. Regardless of the approach, we need to be able to determine what color should be drawn in the intersecting pixels.
3. In your approach, we also want to be able to detect which character is being intersected. Constantin can tell us if our current font support would enable that. I guess your idea is to just draw everything so we would not care. I guess that is OK if there is no alternative.
4. For your approach we will also need to know the exact pixels drawn for the stroked vertical line. I think this is the one that kills the idea since I can't think of a way to get this in Java.

At a minimum, please find the answers to 1 and 2 above. If you can come up with a Java solution for 4 then we can consider this approach, though I still think it is not preferred.

**#45 - 09/30/2015 12:18 PM - Stanislav Lomany**

Java hides the line stroking logic from us. In other words, in Java, how would we know how to set the clip rectangles to overdraw those segments?

If we are going to separate drawing between cells, we'll have to draw strokes manually. Also, I was planning to put locked and scrollable columns on different panels, so manual drawing may really be the way to go.

**#46 - 09/30/2015 12:27 PM - Greg Shah**

I was planning to put locked and scrollable columns on different panels, so manual drawing may really be the way to go.

Our current primitives have no knowledge of the high level panels with which our client operates. The drivers only know about the overall window/canvas being used for drawing.

In other words, you can choose any panel arrangement that makes sense, but the same drawing primitives (e.g. draw a stroked rectangle) will work independently of those panels.

**#47 - 09/30/2015 12:36 PM - Stanislav Lomany**

- *File scroll-error.png added*

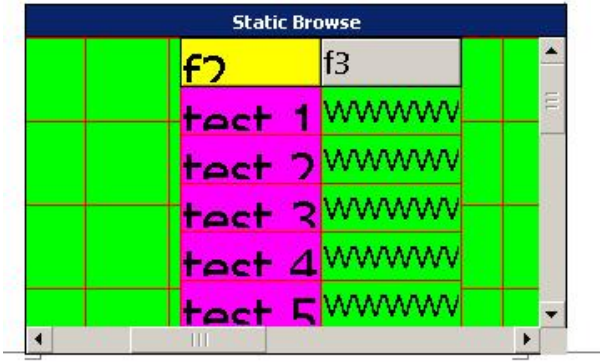
Hynek

Is the change checked in?

2564a

What does it mean "container is scrolled ok", do you see the scroll bars move for example?

See the picture:



I have test grid which is scrolled ok (scroll bar moves correctly), but child columns aren't. However they disappear when they are supposed to go out of the view.

BTW notice drawing artefacts on the bottom.

#48 - 09/30/2015 12:53 PM - Hynek Cihlar

Stanislav Lomany wrote:

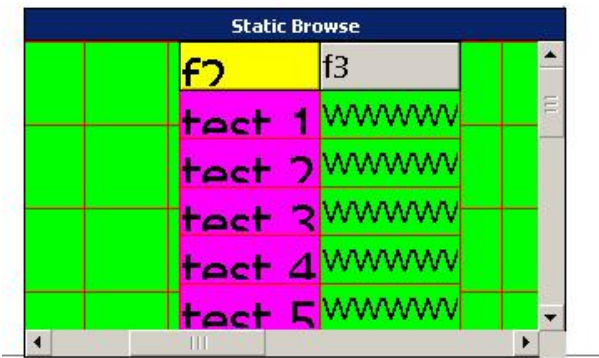
Hynek

Is the change checked in?

2564a

What does it mean "container is scrolled ok", do you see the scroll bars move for example?

See the picture:



I have test grid which is scrolled ok (scroll bar moves correctly), but child columns aren't. However they disappear when they are supposed to go out of the view.

BTW notice drawing artefacts on the bottom.

Please post the code sample you used.

**#49 - 09/30/2015 12:57 PM - Stanislav Lomany**

- File *browse-gui-stat1.p* added

**#50 - 09/30/2015 01:23 PM - Hynek Cihlar**

The problem was in `ColumnSetContainer.draw()`, the `super.draw()` was not properly scoped. See the checked-in fix.

**#51 - 09/30/2015 01:25 PM - Hynek Cihlar**

Btw., I think you should use `NativeScrollContainer` as the base class of `ColumnSetContainer` as your scrolling units are 1:1 to the widget's dimension units.

**#52 - 09/30/2015 01:33 PM - Hynek Cihlar**

Don't worry about the artifacts now, those have been most likely resolved in [#1811](#), task branch 1811r.

**#53 - 09/30/2015 01:38 PM - Stanislav Lomany**

Thank you for your help!

**#54 - 10/01/2015 03:13 AM - Constantin Asofiei**

Greg Shah wrote:

3. In your approach, we also want to be able to detect which character is being intersected. Constantin can tell us if our current font support would enable that. I guess your idea is to just draw everything so we would not care. I guess that is OK if there is no alternative.

To determine which character is intersected, will require measuring the text from left to right and finding which char is intersecting the X where the stroked line needs to be drawn. This gives us the horizontal position, but determining the height of the stroked line where it intersects the character I think is impossible without access to the canvas (there is no way to find the height of the actually drawn pixels, on a certain X position in the drawn character, without drawing the char first and after that measuring the drawn pixels on a certain X; also, what about the case when the stroked pixels in a character are not contiguous, like a G for example - here the intersection might be in the middle, and the stroked line will be "split").

I suspect the BROWSE already is a pretty complex widget to be drawn fully, and each additional drawing operation will have an impact on the Web client. The best way to go is to use the approach mentioned in note 40, as it minimizes both the number of the drawing primitives sent via the WebSocket and processed by the Web client.

**#55 - 10/01/2015 09:41 AM - Stanislav Lomany**



Here is color calculation formula for R, G or B component:

```
inverse = color + 192;   color [0, 64)
inverse = color + 64;   color [64, 128)
inverse = color - 64;   color [128, 192)
inverse = color - 192;  color [192, 255]
```

Note that the colors returned by rdesktop are NOT ACCURATE.

**#56 - 10/01/2015 03:44 PM - Stanislav Lomany**

Guys, about dotted line drawing: so what should I finally do with it - check if is possible to do with Java2D Composite?

**#57 - 10/01/2015 04:10 PM - Greg Shah**

Yes, please investigate if you can match the WIN32 compositing with something from Java2D. If so, this is the preferred option. We will figure out the compositing in the web client, later.

**#58 - 10/02/2015 07:47 AM - Stanislav Lomany**

I have a lucky day: formula from note 55 is actually

```
inverse = color XOR 192
```

And we already have XORComposite in Java2D. I've added GuiDriver.setXORComposite() and GuiDriver.resetComposite() methods which results in desired compositing.

**#59 - 10/02/2015 08:11 AM - Greg Shah**

Fantastic! I assume you have added 2 PaintPrimitive types and the AbstractGuiDriver.setXORComposite()/resetComposite() adds a PaintStructure to the queue?

Leave the GuiWebDriver portion empty. We'll add that part in 1811r.

**#60 - 10/02/2015 08:13 AM - Stanislav Lomany**

I assume you have added 2 PaintPrimitive types and the AbstractGuiDriver.setXORComposite()/resetComposite() adds a PaintStructure to the queue?

Yes.

**#61 - 10/07/2015 01:44 PM - Greg Shah**

Please document the list of open items to be finished before the first pass at GUI browse runtime will be ready for testing. Add an estimate for how long it will take to finish those items.

**#62 - 10/07/2015 02:08 PM - Stanislav Lomany**

1. Right now I'm working on testcases for browse/column sizing/horizontal scrolling.
2. Vertical scrolling with mouse (proper handing of vertical scrolling).
3. Mouse events for cells (selection / current row change).
4. Editable browse.

I'll be able to finish items 1-3 around this weekend. The last one goes ASAP after the previous items. BTW, do we need an editable browse to demonstrate?

**#63 - 10/07/2015 04:38 PM - Greg Shah**

BTW, do we need an editable browse to demonstrate?

I don't think so.

Let's get 2564a into testing after 1-3 are done. Then you can work on editable browse next in branch 2546b.

**#64 - 10/12/2015 02:44 PM - Stanislav Lomany**

Rebased task branch 2564a from P2J trunk revision 10945.

**#65 - 10/12/2015 02:47 PM - Stanislav Lomany**

Guys, after rebase I got couple of problems, one of which confuses me: scrollbars in a scroll pane are not drawn (grey areas instead). Do you have any ideas?

**#66 - 10/12/2015 03:36 PM - Hynek Cihlar**

Please post your test program.

**#67 - 10/12/2015 04:18 PM - Stanislav Lomany**

Hynek, you will have to check out 2564a. Here is the testcase:

```
DEFINE VARIABLE name-hdl AS WIDGET-HANDLE.  
DEFINE VARIABLE num-hdl AS WIDGET-HANDLE.  
DEFINE VARIABLE address-hdl AS WIDGET-HANDLE.  
DEFINE VARIABLE calc-col-hdl AS WIDGET-HANDLE.  
DEFINE VARIABLE browse-hdl AS WIDGET-HANDLE.  
DEFINE VARIABLE buff-field-hdl AS WIDGET-HANDLE.  
DEFINE VARIABLE brws-col-hdl AS WIDGET-HANDLE.  
DEFINE VARIABLE j AS INTEGER.
```

```

def temp-table tt field f1 as integer
                    field f2 as character.

create tt. tt.f1 = 1. tt.f2 = "test 1".
create tt. tt.f1 = 2. tt.f2 = "test 2".

DEFINE FRAME MyFrame WITH SIZE 80 BY 20.
DEFINE QUERY q1 FOR tt SCROLLING.

CREATE BROWSE browse-hdl
    ASSIGN TITLE = "Dynamic Browse"
    X = 2
    Y = 2
    FRAME = FRAME MyFrame:HANDLE
    MULTIPLE = TRUE
    READ-ONLY = YES
    COLUMN-MOVABLE = TRUE
    COLUMN-RESIZABLE = TRUE
    FONT = 10
    QUERY = QUERY q1:HANDLE.

num-hdl = browse-hdl:ADD-LIKE-COLUMN("tt.f1").
name-hdl = browse-hdl:ADD-LIKE-COLUMN("tt.f2").

ENABLE ALL WITH FRAME MyFrame.
OPEN QUERY q1 FOR EACH tt NO-LOCK.
WAIT-FOR CLOSE OF CURRENT-WINDOW.

```

As far as I've checked, scrollbars are not included in the drawing area.

#### #68 - 10/12/2015 06:39 PM - Hynek Cihlar

Stanislav, the problem was in the code initializing the scroll pane. In `BrowseGuiImpl.rebuildColumns` you should not set the scroll bar height/width explicitly, you should leave the scroll pane to do this job. Also, you should inherit `NativeScrollContainer` and not `ScrollContainer`. Do not take assumptions of the scroll bar sizes, again leave this on the scroll pane. Also there is no need to cache the scrollable size, just set the size of `scrollableColumnSet` and leave the scrolling infrastructure to do its job. Hiding of the horizontal scroll bar when the columns don't fit into the viewport is (or should be) also handled automatically.

There are three main parts in the scrolling infrastructure - scroll pane, viewport, scroll widget (or scroll container in your case). Scroll pane holds the

viewport and scrollbars, manages scroll bar visibility, sets viewport and scroll bar sizes, manages border and positions viewport/scroll bars accordingly, routes events. Viewport is the visible portion of the scrollable area, this is what the user is supposed to see. The scroll pane is the whole scrollable area, the effect of scrolling is achieved by changing the location of this pane in respect to the viewport. Changing of the location is handled by NativeScrollContainer automatically in its scroll() method. The visibility of scroll bars is calculated by scroll pane based on the viewport size and total scrollable area and one of the value of the enum ShowBars. Also you should take advantage of the scroll pane's capability to draw border, scroll pane will take care of positioning and sizing of the viewport depending whether border is drawn or not.

#### #69 - 10/12/2015 08:17 PM - Stanislav Lomany

Thank you for your answer! I've put the browse to work and will make adjustments you've recommended ASAP.  
Just for information, there are some new bugs:

1. Because of NPE in AbstractGuiDriver.getBgColor I've moved color calculations from doLayout to draw.
2. I can be wrong, but it seems to me that before rebase I could *hold* down a key to scroll browse. Now only separate presses work.
3. When I leave one of my testcases with Esc, I get this:

```
Caused by: java.lang.NullPointerException: Could not resolve window!
    at com.goldencode.p2j.ui.client.OutputManager.setInvalidate(OutputManager.java:1250)
    at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:13616)
    at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:13515)
    at com.goldencode.p2j.ui.chui.ThinClient.hide(ThinClient.java:6359)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:497)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:705)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:257)
    at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1128)
    at com.goldencode.p2j.net.Queue.transact(Queue.java:585)
    at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:223)
    at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:163)
    at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1425)
    at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:97)
    at com.sun.proxy.$Proxy4.standardEntry(Unknown Source)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:277)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:100)
    at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:201)
    at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:398)
    at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:95)
    at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:267)
```

**#70 - 10/13/2015 04:20 AM - Hynek Cihlar**

- File *BrowseGuiImpl.java.diff* added

Stanislav, see the attached diff. It is not 100% accurate but could help you with the scroll updates. The commented lines should be removed (and in some cases the removal should be compensated in the rest of the layout logic).

**#71 - 10/14/2015 06:50 AM - Stanislav Lomany**

Please review branch 2564a revision 10969.

**#72 - 10/14/2015 08:14 AM - Stanislav Lomany**

The update passed conversion and regression testing.

**#73 - 10/14/2015 08:29 AM - Greg Shah**

Wow! OK, I'll get the code review done ASAP.

**#74 - 10/14/2015 08:32 AM - Greg Shah**

The 2564b changes are all merged into 2272b, right? I think 2564b can be archived.

**#75 - 10/14/2015 09:15 AM - Greg Shah**

Code Review Task Branch 2564a Revision 10969

This is really impressive. Please merge to trunk immediately. The following items can be addressed in branch 2564c as needed.

1. I not sure the `getBrowseCellBottomTextOffset()` belongs in `FontManager`. On the one hand, there is logic that is font-name specific, which we definitely don't like having in widgets. On the other hand, having widget-specific logic in `FontManager` is not good because it is not the place a reader would look to find browse behavior implemented.

Constantin: what are your thoughts here?

2. `BrowseColumnGuiImpl.isFistColumn()` should be `isFirstColumn()`.

**#76 - 10/14/2015 09:18 AM - Stanislav Lomany**

The 2564b changes are all merged into 2272b, right?

Yes.

I think 2564b can be archived.

Done.

**#77 - 10/14/2015 09:28 AM - Greg Shah**

I not sure the `getBrowseCellBottomTextOffset()` belongs in `FontManager`. On the one hand, there is logic that is font-name specific, which we definitely don't like having in widgets. On the other hand, having widget-specific logic in `FontManager` is not good because it is not the place a reader would look to find browse behavior implemented.

Or better yet, is there some font metric that we should be capturing for all fonts that can be reported generically by the `FontManager`, such that the browse-specific logic can be in browse and the font-specific logic is generically supported by the new metric.

**#78 - 10/14/2015 09:29 AM - Greg Shah**

Hynek: do you have any thoughts regarding the issues reported in note 69?

**#79 - 10/14/2015 10:05 AM - Stanislav Lomany**

Hynek: do you have any thoughts regarding the issues reported in note 69?

I've fixed item 3 by changes in browse clean up code.  
Item 1 is not that relevant.  
Item 2 - it affects, say, fill-ins too, so it is a problem.

**#80 - 10/14/2015 10:25 AM - Hynek Cihlar**

Ad item to from note 69.

Stanislav, please check whether key events and/or scroll events are repeated when the key is pressed. Try to stack-trace the calls to `ScrollEvent` ctor, this should tell you how are the key events transformed to scroll events and whether multiple key/scroll events are generated. Use the `LogHelper.dumpStackTrace(String message, int entries)` method for this.

**#81 - 10/14/2015 10:37 AM - Constantin Asofiei**

Greg Shah wrote:

1. I not sure the `getBrowseCellBottomTextOffset()` belongs in `FontManager`. On the one hand, there is logic that is font-name specific, which we definitely don't like having in widgets. On the other hand, having widget-specific logic in `FontManager` is not good because it is not the place a reader would look to find browse behavior implemented.

Constantin: what are your thoughts here?

This might be related to the font's/text's ascent/descent.

Stanislav, can you modify tools/get-font-metrics.p to save the ascent/descent (or maybe other relevant info)? The structure returned by GetTextMetricsA is:

```
typedef struct tagTEXTMETRIC {
    LONG    tmHeight;
    LONG    tmAscent;
    LONG    tmDescent;
    LONG    tmInternalLeading;
    LONG    tmExternalLeading;
    LONG    tmAveCharWidth;
    LONG    tmMaxCharWidth;
    LONG    tmWeight;
    LONG    tmOverhang;
    LONG    tmDigitizedAspectX;
    LONG    tmDigitizedAspectY;
    TCHAR   tmFirstChar;
    TCHAR   tmLastChar;
    TCHAR   tmDefaultChar;
    TCHAR   tmBreakChar;
    BYTE    tmItalic;
    BYTE    tmUnderlined;
    BYTE    tmStruckOut;
    BYTE    tmPitchAndFamily;
    BYTE    tmCharSet;
} TEXTMETRIC
```

You can run this on windev01 via prowini32 -p get-font-metrics.p and specify a font list.

**#82 - 10/14/2015 11:06 AM - Stanislav Lomany**

Task branch 2564a committed into the trunk as bzip revision 10946.

**#83 - 10/14/2015 11:09 AM - Greg Shah**

The following items are high priority:

- font metrics and `FontManager.getBrowseCellBottomTextOffset()` analysis
- editable browse
- key/scroll event repeat issue

What other important features need to be done next?

**#84 - 10/14/2015 11:29 AM - Stanislav Lomany**

Because of constant debugging I was unable to complete original tasks:

0. WIP - random crashes "widget is not attached to the top-level window".
1. WIP - mouse selection.
2. Area to the right of the last column is not clickable (if columns area is smaller than browse width).
3. incorrect value returned by `getVisibleArea` (bug after merge on row selection drawing).
4. implement 4GL GUI quirk that the first row of a single-selection browse is not selected by default;
5. vertical scrolling with mouse (currently keys only);
6. proper horizontal scrolling (currently scrolling with mouse does the job but doesn't match 4GL behavior).
7. column WIDTH support (currently width is based on format width).

**#85 - 10/14/2015 11:44 AM - Greg Shah**

OK, in branch 2564c:

- all the items from note 84
- font metrics and `FontManager.getBrowseCellBottomTextOffset()` analysis/changes
- key/scroll event repeat issue

Then editable browse and the remaining features can be done in subsequent task branches (2564d and beyond).

**#86 - 10/14/2015 11:45 AM - Greg Shah**

I also need to know 2-3 useful GUI browse testcases that we should add to our GUI regression testing list.

**#87 - 10/14/2015 12:36 PM - Stanislav Lomany**

Guys, the issue random crashes "widget is not attached to the top-level window" is actually a bunch of issues caused by the fact that mouse events (like mouse movement) received by widgets (like browse columns or scrollbar buttons) which has no parent window. Most likely these widgets were already destroyed. I haven't seen this issue before recent rebase for 2564a. I don't think I will be able to solve this issue in a reasonable time frame. I'll post the testcase here. Any help appreciated.

**#88 - 10/14/2015 12:38 PM - Hynek Cihlar**

"widget is not attached to the top-level window" will be resolved as part of 2560a. I already have some fixes for this.

**#89 - 10/14/2015 12:39 PM - Hynek Cihlar**



Stanislav Lomany wrote:

I'll post the testcase here.

Please do.

**#90 - 10/14/2015 12:54 PM - Stanislav Lomany**

- File *go.p* added

- File *brws-size-implicit-dyn.p* added

Reproduction:

1. Run *go.p*.
2. Run *brws-size-implicit-dyn* from it.
3. Click on the right arrow of the bottom scrollbar.
4. ESC twice to return to *go.p*.
5. Run *brws-size-implicit-dyn* again.
6. Position mouse over the browse.

At this point there should be a bend. If not - clicking on the browse and returning to *go.p* and back finally allows to get it.

**#91 - 10/15/2015 07:32 AM - Stanislav Lomany**

Current status:

0. random crashes "widget is not attached to the top-level window" - hope that Hynek works on that.
1. Mouse selection - **implemented**. However I left two issues for **later implementation**: 1A combinations with ctrl and shift keys and 1B clicking very close to the edge of a row will result in selection of another row.
2. Area to the right of the last column is not clickable (if columns area is smaller than browse width) - **fixed**.
3. incorrect value returned by *getVisibleArea* (bug after merge on row selection drawing) - **fixed** (in *ScrollPaneGuiImpl*).
4. implement 4GL GUI quirk that the first row of a single-selection browse is not selected by default - **postponed**: row selection logic is deeply intergated in UI-independent code, so it has to be done carefully, while impact of this issue on functionality is not that significant.
5. vertical scrolling with mouse (currently keys only) - **WIP**

**#92 - 10/15/2015 07:35 AM - Stanislav Lomany**

BTW when do you expect to have a look how browse works in production/demonstration?

**#93 - 10/15/2015 08:31 AM - Greg Shah**

when do you expect to have a look how browse works in production/demonstration?

As soon as 2560a is merged to trunk (and 2272b is rebased) we will be able to give it a try in the 2272b branch.

We know that there are going to be some important fixes in 1811s and in 2564c.

All 4 branches will be important to get merged to trunk for the testcases to be expected to work. At that point we will need to test the demo scenarios, then identify and fix any issues.

#### #94 - 10/15/2015 06:21 PM - Stanislav Lomany

Current status:

5. vertical scrolling with mouse (currently keys only) - **implemented** (still need to revisit it in the future to check the behavior for row deletion).
6. proper horizontal scrolling (currently scrolling with mouse does the job but doesn't match 4GL behavior) - **started**.

#### #95 - 10/16/2015 09:43 AM - Stanislav Lomany

Found additional issue related to item 5: keys misbehavior when the current row is out of the view.

#### #96 - 10/16/2015 02:27 PM - Constantin Asofiei

Stanislav, the abend in 1811 note 1516 is caused by the fact that `BrowseGuiImpl.rebuildColumns` creates a scrollpane, adds it to the tree, but it doesn't add it to the registry... so it can't be found by `TC.getWidget`. Is there a reason why this is not happening?

The Web driver needs to register all mouse-aware widgets (like scrollbars, scroll buttons, etc) to know when to send the event to the Java side.

Also, `BrowseGuiImpl` should have an overridden `mouseActions()` method, with the mouse actions supported by the widget. I think only `MOUSE-CLICK` is needed at this time.

#### #97 - 10/16/2015 02:43 PM - Stanislav Lomany

Constantin, consider the components:

```
Browse
  Title panel
  Column set 1
    Browse column 1
  Scroll panel
    Column set 2
      Browse column 2
      Browse column 3
  Scroll bars
```

Column sets are registered using

```
OutputManager.instance().getRegistry().addWidget((Widget) this);
```

And unregistered on destroy. (This is not checked in trunk yet.)

Should I make the same thing for scroll panel? Title panel?

Also, `BrowseGuiImpl` should have an overridden `mouseActions()` method, with the mouse actions supported by the widget.

I have mouseClicked overridden. Isn't that enough?

**#98 - 10/16/2015 02:50 PM - Constantin Asofiei**

Stanislav Lomany wrote:

Constantin, consider the components:

[...]

Column sets are registered using

[...]

And unregistered on destroy. (This is not checked in trunk yet.)

Should I make the same thing for scroll panel? Title panel?

Yes, do this for both.

Also, BrowseGuiImpl should have an overridden mouseActions() method, with the mouse actions supported by the widget.

I have mouseClicked overridden. Isn't that enough?

No, is not enough. Please add this method to BrowseGuiImpl:

```
protected int[] mouseActions()
{
    return new int[]
    {
        MouseEvent.MOUSE_CLICKED,
    };
}
```

#### #99 - 10/16/2015 02:53 PM - Hynek Cihlar

Stanislav, before I forget. I noticed that you resolve the main window in `BrowseGuiImpl` and `BrowseColumnGuiImpl` through `Window.resolveWindow()`. Is this correct? `Window.resolveWindow()` may give you a different window than the one the widget is attached to, for example when the browse's actual window is hidden.

#### #100 - 10/16/2015 02:55 PM - Constantin Asofiei

Hynek Cihlar wrote:

Stanislav, before I forget. I noticed that you resolve the main window in `BrowseGuiImpl` and `BrowseColumnGuiImpl` through `Window.resolveWindow()`. Is this correct? `Window.resolveWindow()` may give you a different window than the one the widget is attached to, for example when the browse's actual window is hidden.

No, is not correct, always use `AbstractWidget.window()` to resolve the widget's window.

#### #101 - 10/16/2015 03:58 PM - Stanislav Lomany

Does usage of this function seems valid to you, guys? `Window` may not be available in `afterConfigUpdate`.

```
public Window window()
{
    Window window = parent(Window.class);

    if (window == null && (config.frameId != -1))
    {
        window = ThinClient.getInstance().getFrame(config.frameId).window();
    }

    return window;
}
```

#### #102 - 10/16/2015 04:45 PM - Hynek Cihlar

Stanislav Lomany wrote:

`Window` may not be available in `afterConfigUpdate`.

Yes, this is true. Just don't call `window()` before the widget is attached to its parent and to its main window. Don't cache the window reference in the browse classes, directly call `window()` when needed.

**#103 - 10/17/2015 07:13 PM - Stanislav Lomany**

In order to lay out parent frame properly I have to explicitly set browse dimensions even if browse size is implicit (default or DOWN is used). In order to calculate browse size, I need font metrics, which require window reference. So I'm intended to use the function posted in note 101. Let me know if it can potentially ruin something.

**#104 - 10/18/2015 05:28 AM - Hynek Cihlar**

Stanislav Lomany wrote:

In order to lay out parent frame properly I have to explicitly set browse dimensions even if browse size is implicit (default or DOWN is used). In order to calculate browse size, I need font metrics, which require window reference. So I'm intended to use the function posted in note 101. Let me know if it can potentially ruin something.

You always require the reference of the window the browse widget is attached to. So yes, window() is what you need. Just make sure the widget is already attached to the widget tree before you call window().

**#105 - 10/18/2015 05:36 AM - Stanislav Lomany**

Hynek, just to make sure that we are on the same page. The overridden window() impelmentation posted in note 101 is OK, right? I'm checking the returned value for null of course.

**#106 - 10/18/2015 11:36 AM - Hynek Cihlar**

Stanislav Lomany wrote:

Hynek, just to make sure that we are on the same page. The overridden window() impelmentation posted in note 101 is OK, right?

Yes.

**#107 - 10/19/2015 08:10 AM - Greg Shah**

Please summarize the status of 2564c. Based on the code that is there, it seems you are done or almost done with the anticipated items.

Is the code ready for review?

**#108 - 10/19/2015 08:42 AM - Stanislav Lomany**

Finished horizontal scrolling and column's width. Right now I'm trying to make browse layout more correctly - once in proper place. How much time do I have?

**#109 - 10/19/2015 09:24 AM - Greg Shah**

How much time do I have?

We need to merge to trunk today, so that means code review and testing in 3 hours.

**#110 - 10/19/2015 11:19 AM - Greg Shah**

2560a is going to be merged to trunk in the next 20 minutes. When that happens, please make sure you rebase before you post your final code.

**#111 - 10/19/2015 12:02 PM - Stanislav Lomany**

OK. I'm writing comments and will rebase and check the behavior after that.

**#112 - 10/19/2015 02:39 PM - Greg Shah**

What is the status on the rebase and the final code?

**#113 - 10/19/2015 02:40 PM - Stanislav Lomany**

I had to fix minor issue which seemed important to me. I'm finishing comments and let you know when rebase is done.

**#114 - 10/19/2015 03:44 PM - Stanislav Lomany**

Rebased task branch 2564c from P2J trunk revision 10948.  
Please review this branch rev 10958. No regressions after rebase.

**#115 - 10/19/2015 04:02 PM - Greg Shah**

Code Review Task Branch 2564c Revision 10958

I don't see any problems.

Hynek: would you please review the scrolling-related changes?

**#116 - 10/19/2015 04:56 PM - Hynek Cihlar**

Code Review Task Branch 2564c Revision 10958.

This version is a great improvement from the last version I saw. I have some objections though.

Stanislav, ideally you shouldn't hold the references to scroll bars in your widgets. This is probably related to the new methods `setIgnorePanelSize()` and `setSendNonTrimmedEvents()` you created. The javadoc for `setSendNonTrimmedEvents()` says, "This is useful to catch intention to scroll out of the current portion of the result set." Can you explain why is the "non-trimmed" scrolling needed and the use case behind this?

Also, scrolling the scrollbars shouldn't be done by calling `setPosition()` on the scroll bar references. You added the `ScrollPaneGuiImpl.setManualPositionControl()` for this purpose. What is the use case here? Why isn't the standard mechanism of notifying the scroll pane sufficient?

**#117 - 10/19/2015 05:27 PM - Stanislav Lomany**

Can you explain why is the "non-trimmed" scrolling needed and the use case behind this?

It is used for scrolling rows. There is the estimated number of rows (max). If user scrolls beyond max, we load a new portion of rows and update max.

Also, scrolling the scrollbars shouldn't be done by calling `setPosition()` on the scroll bar references. You added the `ScrollPaneGuiImpl.setManualPositionControl()` for this purpose. What is the use case here? Why isn't the standard mechanism of notifying the scroll pane sufficient?

Vertical scrolling is not related to the scrolled panel size or position at all. For horizontal scrolling also has COLUMN-SCROLLING mode which has fixed number of positions which match column count and not related to width of columns.

**#118 - 10/19/2015 05:48 PM - Hynek Cihlar**

Stanislav Lomany wrote:

Can you explain why is the "non-trimmed" scrolling needed and the use case behind this?

It is used for scrolling rows. There is the estimated number of rows (max). If user scrolls beyond max, we load a new portion of rows and update max.

I see. I believe it should be enough to override `NativeScrollContainer.getScrollDimension()` and return the actual dimensions according to the number of loaded rows. Scroll pane should update scroll bars accordingly.

Also, scrolling the scrollbars shouldn't be done by calling `setPosition()` on the scroll bar references. You added the `ScrollPaneGuiImpl.setManualPositionControl()` for this purpose. What is the use case here? Why isn't the standard mechanism of notifying the scroll pane sufficient?

Vertical scrolling is not related to the scrolled panel size or position at all. For horizontal scrolling also has COLUMN-SCROLLING mode which has fixed number of positions which match column count and not related to width of columns.

The unit of measure for the scroll position and scroll range returned by `ScrollableWidget.getScrollDimension()`, `ScrollableWidget.getVisibleDimension()` and `ScrollableWidget.getScrollStep()` is arbitrary. You can even choose different units for vertical and horizontal scrolling. Again, it is sufficient to override these methods in `ColumnSetContainer` and return the columns instead of sizes.

In the end you shouldn't require accessing the scroll bars directly.

**#119 - 10/19/2015 06:11 PM - Stanislav Lomany**

I see. I believe it should be enough to override `NativeScrollContainer.getScrollDimension()` and return the actual dimensions according to the number of loaded rows. Scroll pane should update scroll bars accordingly.

I don't think so. I need an event which indicates that we are beyond max. All current events are trimmed to max and I cannot tell if it is max or max + 1 position and return new value in `getScrollDimension()`.

Again, it is sufficient to override these methods in `ColumnSetContainer` and return the columns instead of sizes.

Well, I believe it is so. But I have concerns in terms of simplicity and incapsulation. I will need to implement these function so the calculations in second case will end up in required behavior, while all I need is the first formula.

```
if (ignorePanelSize)
{
    tHeight = Coordinate.scale(travelSize * getStep() / getMax());
}
else
{
    int visibleRange = viewport.getScrollWidget().getVisibleDimension().height;
    double unitHeight = (double) d.height / visibleRange;

    // draw scroll bar thumb
    int totalHeight = (int)
        (viewport.getScrollWidget().getScrollDimension().height * unitHeight);
    tHeight = Coordinate.scale(travelSize * d.height / totalHeight);
}
```

**#120 - 10/19/2015 06:30 PM - Hynek Cihlar**

Stanislav Lomany wrote:



I see. I believe it should be enough to override `NativeScrollContainer.getScrollDimension()` and return the actual dimensions according to the number of loaded rows. Scroll pane should update scroll bars accordingly.

I don't think so. I need an event which indicates that we are beyond max. All current events are trimmed to max and I cannot tell if it is max or max + 1 position and return new value in `getScrollDimension()`.

I see, you need to load the rows not when you reach max, but when at max and the user clicks on the scrollbar.

It would be cleaner to let scroll pane notify scrollable widget (`ColumnSetContainer`) about the scroll request. New method `ScrollableWidget.scrollRequest(oldPosition, newPosition)` would allow the scrollable widget to readjust the max value. Without the need to reference scroll bars directly and implementing the specific use cases in scroll pane.

Again, it is sufficient to override these methods in `ColumnSetContainer` and return the columns instead of sizes.

Well, I believe it is so. But I have concerns in terms of simplicity and incapsulation. I will need to implement these function so the calculations in second case will end up in required behavior, while all I need is the first formula.  
[...]

Yes, but we need the second formula anyway. So what is more complicated?

```
if (ignorePanelSize)
{
    tHeight = Coordinate.scale(travelSize * getStep() / getMax());
}
else
{
    int visibleRange = viewport.getScrollWidget().getVisibleDimension().height;
    double unitHeight = (double) d.height / visibleRange;

    // draw scroll bar thumb
    int totalHeight = (int)
        (viewport.getScrollWidget().getScrollDimension().height * unitHeight);
    tHeight = Coordinate.scale(travelSize * d.height / totalHeight);
}
```

or

```
int visibleRange = viewport.getScrollWidget().getVisibleDimension().height;
double unitHeight = (double) d.height / visibleRange;

// draw scroll bar thumb
int totalHeight = (int)
    (viewport.getScrollWidget().getScrollDimension().height * unitHeight);
tHeight = Coordinate.scale(travelSize * d.height / totalHeight);
```

?

**#121 - 10/19/2015 06:59 PM - Stanislav Lomany**

First issue: why not add handleScrollRequest method to ScrollListener?

Second issue: I'm not sure that your solution is possible (if I understand it correctly): I cannot affect `viewport.getScrollWidget().getVisibleDimension().height` and `viewport.getScrollWidget().getScrollDimension().height` because `getScrollWidget()` returns the panel and not some class designed to solve scrolling problem.

**#122 - 10/19/2015 07:30 PM - Hynek Cihlar**

Stanislav Lomany wrote:

First issue: why not add handleScrollRequest method to ScrollListener?

I would rather not add the method to ScrollListener. Scrollable widgets don't implement ScrollListener, they don't and should not process scroll events directly, they only generate scroll events. Scroll events should be processed by scroll pane only, it knows how to route them to all participating components.

Second issue: I'm not sure that your solution is possible (if I understand it correctly): I cannot affect `viewport.getScrollWidget().getVisibleDimension().height` and `viewport.getScrollWidget().getScrollDimension().height` because `getScrollWidget()` returns the panel and not some class designed to solve scrolling problem.

`viewport.getScrollWidget()` returns ScrollableWidget or indirectly ColumnSetContainer which is completely under your control.

**#123 - 10/19/2015 07:41 PM - Stanislav Lomany**

`viewport.getScrollWidget()` returns ScrollableWidget or indirectly ColumnSetContainer which is completely under your control.

Right, but I expect `ColumnSetContainer.getVisibleDimension()` and `ColumnSetContainer.getScrollDimension()` to return actual dimensions. And `getScrollDimension()/getVisibleDimension()` ratio differs from what I want to get from the scrollbar.

**#124 - 10/19/2015 07:49 PM - Hynek Cihlar**

Stanislav Lomany wrote:

`viewport.getScrollWidget()` returns `ScrollableWidget` or indirectly `ColumnSetContainer` which is completely under your control.

Right, but I expect `ColumnSetContainer.getVisibleDimension()` and `ColumnSetContainer.getScrollDimension()` to return actual dimensions. And `getScrollDimension()/getVisibleDimension()` ratio differs from what I want to get from the scrollbar.

Scroll dimension doesn't have to correspond with the widget's physical dimensions. See the javadoc of `ScrollableWidget.getScrollDimension()`:

```
* The returned dimension doesn't need to be in the same units as the
* widget's physical size but it must be compatible with the result
* of {@link #getVisibleDimension()} and {@link #getScrollStep()}.
```

Also see how for example lists implement the scrollable widgets (`DefaultList`), they also don't operate in physical dimensions, but line numbers.

**#125 - 10/20/2015 08:30 AM - Stanislav Lomany**

Also, scrolling the scrollbars shouldn't be done by calling `setPosition()` on the scroll bar references.

How should I update scrollbar position if, say, browse was scrolled by keys?

**#126 - 10/20/2015 08:48 AM - Hynek Cihlar**

Stanislav Lomany wrote:

Also, scrolling the scrollbars shouldn't be done by calling `setPosition()` on the scroll bar references.

How should I update scrollbar position if, say, browse was scrolled by keys?

ScrollPane moves the scroll bars. You only notify it with a scroll event, use ScrollContainer.postScrollEvent() for this. See EditorGuiImpl.mouseWheelMoved() for an example.

**#127 - 10/20/2015 12:36 PM - Stanislav Lomany**

Can I update max value with setMax() when it is nessecary?

**#128 - 10/20/2015 01:11 PM - Stanislav Lomany**

Since scrollbar changes are GUI-only, I successfully ran regression testing (conversion and runtime).

**#129 - 10/20/2015 01:18 PM - Hynek Cihlar**

Stanislav Lomany wrote:

Can I update max value with setMax() when it is nessecary?

You shouldn't do this directly, again this is up to the scroll pane. What is the use case?

**#130 - 10/20/2015 01:40 PM - Greg Shah**

Since scrollbar changes are GUI-only, I successfully ran regression testing (conversion and runtime).

Excellent! There won't be a need to rerun regression testing in 2564c directly.

When the scrolling changes are agreed and completed, please merge all the 2564c changes into 1811s.

**#131 - 10/20/2015 01:46 PM - Stanislav Lomany**

OK, we have:

Scroll dimension

Visible dimension - fixed value

Max = Scroll dimension - Visible dimension

And this value is assigned upon scrollbar initiation:

Scroll step

Are you suggesting to update Scroll dimension as the data set grows? In this case we need to explicitly update Scroll step = Max / Number of rows.

**#132 - 10/20/2015 01:52 PM - Hynek Cihlar**

Stanislav Lomany wrote:

OK, we have:  
Scroll dimension  
Visible dimension - fixed value  
Max = Scroll dimension - Visible dimension

And this value is assigned upon scrollbar initiation:  
Scroll step

Are you suggesting to update Scroll dimension as the data set grows? In this case we need to explicitly update Scroll step = Max / Number of rows.

Please modify scroll pane to update step as necessary. I think it should be enough to call `bar.setStep()` from `ScrollPaneGuiImpl.adjustScrollLayout()`.

**#133 - 10/20/2015 03:49 PM - Stanislav Lomany**

Scroll dimension = Visible dimension \* Number of rows / DOWN.  
Scroll step = (Scroll dimension - Visible dimension) / (Number of rows - DOWN)

Step and position are actually double values, not integer.

**#134 - 10/20/2015 04:05 PM - Hynek Cihlar**

Stanislav Lomany wrote:

Scroll dimension = Visible dimension \* Number of rows / DOWN.  
Scroll step = (Scroll dimension - Visible dimension) / (Number of rows - DOWN)

Step and position are actually double values, not integer.

Shouldn't be step constant?

**#135 - 10/20/2015 04:12 PM - Stanislav Lomany**

It it will be constant then we will not be able to affect thumb size.

**#136 - 10/20/2015 04:17 PM - Hynek Cihlar**

Stanislav Lomany wrote:

It it will be constant then we will not be able to affect thumb size.

Thumb size is not affected by step, see `ScrollBarGuiImpl.drawScrollBarThumb()`.

**#137 - 10/20/2015 04:37 PM - Stanislav Lomany**

Thumb size is affected by Scroll dimension and Visible dimension. And they are linked in this equation:

$$\text{Scroll step} = (\text{Scroll dimension} - \text{Visible dimension}) / (\text{Number of rows} - \text{DOWN})$$

in which I cannot vary Visible dimension, Number of rows and DOWN.

**#138 - 10/20/2015 04:49 PM - Hynek Cihlar**

Stanislav Lomany wrote:

Thumb size is affected by Scroll dimension and Visible dimension. And they are linked in this equation:

$$\text{Scroll step} = (\text{Scroll dimension} - \text{Visible dimension}) / (\text{Number of rows} - \text{DOWN})$$

in which I cannot vary Visible dimension, Number of rows and DOWN.

I would expect the step to be equal to the height of a row or a multiple. How did you come up with this equation?

**#139 - 10/20/2015 05:50 PM - Stanislav Lomany**

Scroll step is height to be scrolled / number of rows in this height.

**#140 - 10/20/2015 05:52 PM - Stanislav Lomany**

"height to be scrolled" is actually max.

**#141 - 10/20/2015 07:46 PM - Stanislav Lomany**

Guys, here is basic implementation for vertical scrolling. Let me know your thoughts.

First, I wrote this function for reliability and to avoid recursive invocations:

```
private NativeDimension getViewportDimension()
{
    int width = physicalDimension().width - ScrollBarGuiImpl.SCROLLBAR_SIZE -
                (config.box ? 2 * BORDER_THICKNESS : 0);
    int height = physicalDimension().height - (config.box ? 2 * BORDER_THICKNESS : 0);
    if (hasTitle())
    {
        height -= titleHeight + BORDER_THICKNESS;
    }
    if (totalColumnsWidth > width)
```

```

    {
        height -= ScrollBarGuiImpl.SCROLLBAR_SIZE;
    }
    return new NativeDimension(width, height);
}

```

#### Functions in ScrollableColumnSet:

```

@Override
public void scroll(Integer newXPosition, Integer newYPosition)
{
    int newRow = newYPosition / getScrollStep().height; // is that reliable?
    if (getCacheSize() > 0)
    {
        int oldRow = getCachedRowNumber(0);
        if (oldRow == newRow)
        {
            return;
        }

        incrementalScroll(oldRow < newRow ? 1 : -1);
        repaint();
    }

    super.scroll(newXPosition, physicalLocation.y); // avoid actual vertical panel scrolling
}

@Override
public NativeDimension getScrollStep()
{
    int vStep = 1;
    if (BrowseGuiImpl.this.config.maxDataGuess > getDown())
    {
        vStep = (getScrollDimension().height - getVisibleDimension().height) /
            (BrowseGuiImpl.this.config.maxDataGuess - getDown());
    }

    return new NativeDimension(1, vStep);
}

@Override
public NativeDimension getVisibleDimension()
{
    return getViewPortDimension();
}

@Override
public NativeDimension getScrollDimension()
{
    NativeDimension res = getVisibleDimension();
    if (getDown() > 0)
    {
        res = new NativeDimension(res.width, res.height * BrowseGuiImpl.this.config.maxDataGuess / getDown());
    }
    return res;
}

public void notifyBrowseScrolled()
{
    postScrollEvent(null, getCachedRowNumber(0) * getScrollStep().height);
}

```

Haven't checked edge cases yet.

**#142 - 10/21/2015 04:27 AM - Hynek Cihlar**

Stanislav Lomany wrote:

Scroll step is height to be scrolled / number of rows in this height.

This is effectively a height of one row, a constant. Just make sure you return this value in `ColumnSetContainer.getScrollStep()`.

**#143 - 10/21/2015 04:53 AM - Hynek Cihlar**

Stanislav Lomany wrote:

Guys, here is basic implementation for vertical scrolling. Let me know your thoughts.

First, I wrote this function for reliability and to avoid recursive invocations:

[...]

Functions in `ScrollableColumnSet`:

[...]

Haven't checked edge cases yet.

You shouldn't override `getVisibleDimension()`, `NativeScrollContainer.getVisibleDimension()` already takes care of calculating the correct viewport size.

Are the units for the horizontal scroll pixels or columns? If columns, you should return number of columns in `getScrollDimension()` and in `scroll()` convert `newXPosition` to pixel position before calling `setPhysicalLocation()`.

In `scroll()`, do not call `super.scroll()` but set the calculated location with `setPhysicalLocation()` and `repaint()`.

I am afraid am not getting how you calculate the vertical scroll size (which doesn't mean it shouldn't be correct :-)). But I would expect the following equation:  $\text{row-height} * \text{number-of-loaded-rows}$ . Where `row-height` is calculated as font height plus some margin.

**#144 - 10/21/2015 05:17 AM - Stanislav Lomany**



You shouldn't override `getVisibleDimension()`, `NativeScrollContainer.getVisibleDimension()` already takes care of calculating the correct viewport size.

`ScrollPaneGuiImpl.calcViewportSize()` calculates visible dimension basing on scroll dimension. And we are calculation scroll dimension basing on visible dimension.

#### #145 - 10/21/2015 05:37 AM - Hynek Cihlar

Stanislav Lomany wrote:

You shouldn't override `getVisibleDimension()`, `NativeScrollContainer.getVisibleDimension()` already takes care of calculating the correct viewport size.

`ScrollPaneGuiImpl.calcViewportSize()` calculates visible dimension basing on scroll dimension. And we are calculation scroll dimension basing on visible dimension.

It needs to know the scrollable dimension to figure out whether scroll bars should be displayed or not. Just return the correct dimension from `ScrollableColumnSet.getScrollDimension()` and it will calculate the viewport size and the scroll layout for you.

#### #146 - 10/21/2015 06:11 AM - Stanislav Lomany

I suggest to save time to both of us. There is the single testcase that works as in 4GL. I'll check it the current code version and can you please make modifications that you find appropriate?

#### #147 - 10/21/2015 06:36 AM - Stanislav Lomany

- File *brws-size-chars.png* added

Testcase: `uast/browse/gui/brws-size-chars.p`

My font table: `uast/browse/gui/fonts-table.txt`

Screenshot how it looks on my side (after it has been scrolled to the bottom and then to the top) is attached.

Checked in rev 10959 of 2564c.

If you need it - `rowHeight` holds row height. But if you have temptation to set it as the step, make sure that the thumb height is correct.

Estimated number of rows is in `config.maxDataGuess`, it is 100 at the beginning until we have scrolled the actual end of row set and it turns to 8 in our case.

#148 - 10/21/2015 07:18 AM - Hynek Cihlar

Stanislav Lomany wrote:

I suggest to save time to both of us. There is the single testcase that works as in 4GL. I'll check it the current code version and can you please make modifications that you find appropriate?

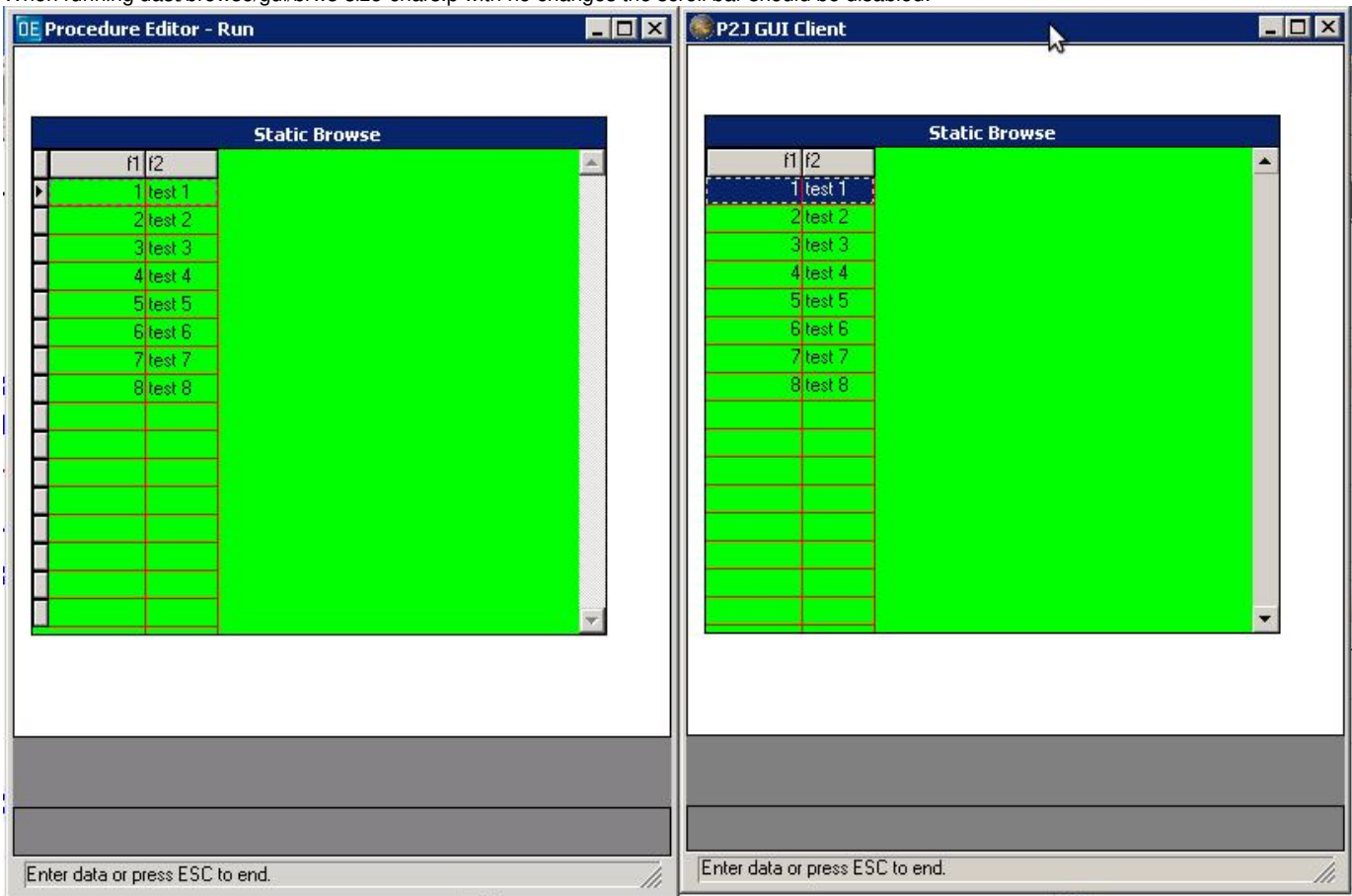
Sure, I will look at it.

#149 - 10/21/2015 12:04 PM - Hynek Cihlar

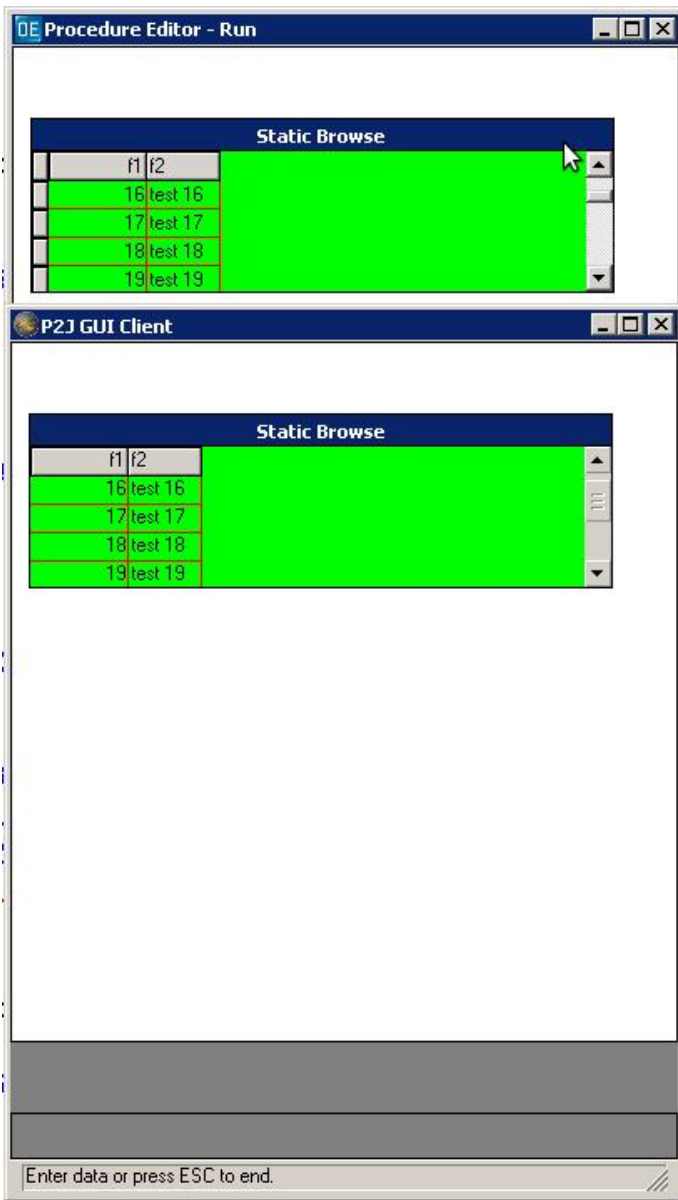
- File `browse_scroll_20_db_entries_5_height.png` added

Stanislav I am testing 10959 of 2564c and found some differences between P2J and 4GL.

When running `uast/browse/gui/brws-size-chars.p` with no changes the scroll bar should be disabled:



When running `uast/browse/gui/brws-size-chars.p` with 20 DB entries and height of 5 chars the thumb size is different.



It looks like there are two scrolling modes. First when it doesn't know how many total result entries there is in the supplied query and second when you hit the end. In the first mode it somehow calculates the thumb size (do we know how?) and when you scroll down you reach the last entry before the thumb hits the lower button. In the second mode it adjusts the thumb size and the scroll thumb travel length spans the whole scroll bar height - when you scroll to the last entry the thumb hits the lower button.

Stanislav you also mentioned something about paging, the case with max + 1. When user is at max position and he clicks on the lower scroll button, the browse loads more entries. Do I interpret this correctly? How do I test it?

Greg, do we need to spend time on the correct thumb size now or can we defer this for later?

**#150 - 10/21/2015 12:06 PM - Hynek Cihlar**

- *File browse\_scroll\_8\_db\_entries\_15\_height.png added*

**#151 - 10/21/2015 12:15 PM - Greg Shah**

Some thoughts/questions:

1. What is the status of your concerns from note 116? It is not clear to me which of your concerns are still open to further cleanup.
2. Are the only open issues related to the scroll thumb size and paging?
3. 1811s has some improvements that fix dragging the scrollbar thumb button and also fix clicking in the non-thumb area. In addition, there are other drawing cleanups for scrollbars. It may make sense to merge with 1811s sooner rather than later. Additional fixes can be put in 1811s, if we are quick about it.

Greg, do we need to spend time on the correct thumb size now or can we defer this for later?

The 2nd and 3rd demo scenarios definitely have browse instances where the scrollbars are not disabled. To get them matching sufficiently, we will need to solve the thumb size.

But we can at least get everything else merged into 1811s, so that when 2272b is merged to trunk, we will be able to rebase and for the first time actually work on the demo scenarios.

**#152 - 10/21/2015 12:34 PM - Hynek Cihlar**

Greg Shah wrote:

Some thoughts/questions:

1. What is the status of your concerns from note 116? It is not clear to me which of your concerns are still open to further cleanup.

The concerns mentioned in note 116 are still open. But we can defer any cleanup for later I think.

2. Are the only open issues related to the scroll thumb size and paging?

There is also the issue with the scroll bar not being disabled when all entries fit on one page.

**#153 - 10/21/2015 12:41 PM - Stanislav Lomany**

When running `uast/browse/gui/brws-size-chars.p` with no changes the scroll bar should be disabled

I'm aware of that. It worked fine in 10958. 10959 is a kind of concept.

Stanislav you also mentioned something about paging, the case with `max + 1`. When user is at max position and he clicks on the lower scroll button, the browse loads more entries. Do I interpret this correctly? How do I test it?

Create more than 100 records.

Hynek, please leave scrollbar sizing stuff to me. My main problem is that I cannot bring scrollbar logic to the state that is correct to you. It would be enough from your side to leave it in the state "somewhat similar to 4GL". And still don't get why I cannot use scrollbar manually (i.e. set max, position, enabled and some thumb size ratio / min size) and should try to fit it in the concert of panels instead :)

**#154 - 10/21/2015 12:49 PM - Greg Shah**

I don't want to leave 10959 in a partial state. If we can make changes (ignoring the thumb size issues) to address the note 116 issues, let's do so.

We are about to rebase 1811s. I prefer to do this after we merge 2564c into 1811s, as this will eliminate the need to rebase 2564c (and I think there are some conflicts to deal with). This is a high priority as it will unblock our ability to test the demo scenarios in 1811s.

We need to quickly come to a conclusion of 10959 and the merge to 1811s.

**#155 - 10/21/2015 01:06 PM - Hynek Cihlar**

Greg Shah wrote:

I don't want to leave 10959 in a partial state. If we can make changes (ignoring the thumb size issues) to address the note 116 issues, let's do so.

We are about to rebase 1811s. I prefer to do this after we merge 2564c into 1811s, as this will eliminate the need to rebase 2564c (and I think there are some conflicts to deal with). This is a high priority as it will unblock our ability to test the demo scenarios in 1811s.

We need to quickly come to a conclusion of 10959 and the merge to 1811s.

I think the easiest would be to merge to 1811s and deal with the scroll issues later.

**#156 - 10/21/2015 01:08 PM - Stanislav Lomany**

My vote goes for this: 1. Commit 10958. 2. Check thumb size behavior found by Hynek. 3. Discuss the future changes with him when time permits.

**#157 - 10/21/2015 01:12 PM - Stanislav Lomany**

I think the easiest would be to merge to 1811s and deal with the scroll issues later.

Do you mean 10958 or 10959?

**#158 - 10/21/2015 01:20 PM - Hynek Cihlar**

Stanislav Lomany wrote:

I think the easiest would be to merge to 1811s and deal with the scroll issues later.

Do you mean 10958 or 10959?

If there are no functional improvements in 10959 then I would merge 10958.

**#159 - 10/21/2015 01:21 PM - Greg Shah**

OK, go ahead with the merge. When you are done, I will rebase 1811s.

**#160 - 10/21/2015 01:25 PM - Stanislav Lomany**

Should I merge it into trunk?

**#161 - 10/21/2015 01:27 PM - Greg Shah**

I was assuming you would merge it into 1811s.

But since you have already passed regression testing, I guess it is OK to merge to trunk. It will require you to rebase from 10949 first and then do the merge.

Tell me what you prefer.

**#162 - 10/21/2015 01:38 PM - Stanislav Lomany**

There are some conflicting files with trunk, so I would rather merge to 1811s. Should it be merged with "bzt merge" or just changed commit files?

**#163 - 10/21/2015 01:43 PM - Greg Shah**

Use bzt merge so that the history is maintained properly.

**#164 - 10/21/2015 02:21 PM - Stanislav Lomany**

2564c merged into 1811s as rev 11008.

**#165 - 10/21/2015 05:51 PM - Stanislav Lomany**

Practice showed that I set that bracket wrong - scrollbars are disabled. Will address in the nearest update.

**#166 - 10/22/2015 03:40 AM - Stanislav Lomany**

Thumb size is  $\text{scale}(\text{Travel size} * \text{DOWN} / \text{Estimated rows})$ , minimum 8.

Thumb position is  $\text{min}(\text{Travel size} - \text{Thumb size}, \text{scale}((\text{Travel size} - \text{Thumb size}) * 0\text{-based row index} / (\text{Estimated rows} - \text{DOWN})))$ .

There are no two scrolling modes, it is all about the number of estimated rows.

**#167 - 10/22/2015 04:28 AM - Hynek Cihlar**

Stanislav Lomany wrote:

Thumb size is  $\text{scale}(\text{Travel size} * \text{DOWN} / \text{Estimated rows})$ , minimum 8.

Thumb position is  $\text{min}(\text{Travel size} - \text{Thumb size}, \text{scale}((\text{Travel size} - \text{Thumb size}) * 0\text{-based row index} / (\text{Estimated rows} - \text{DOWN})))$ .

There are no two scrolling modes, it is all about the number of estimated rows.

Stanislav, of course it is always about the number of rows and range. But functionally, from the users point of view, it behaves differently when you do a first pass scroll and subsequent passes.

**#168 - 10/22/2015 04:34 AM - Stanislav Lomany**

Good morning! How do you think, if we control scrolling using Visible dimension and Scroll dimension, are we able to implement this behavior?

**#169 - 10/22/2015 05:06 AM - Hynek Cihlar**

Stanislav Lomany wrote:

Good morning! How do you think, if we control scrolling using Visible dimension and Scroll dimension, are we able to implement this behavior?

Yes, I think so.

Regarding the thumb size. 4GL uses the native win32 scroll control. It doesn't allow to explicitly set the thumb size. The control calculates the size based on actual range and physical size of the bar. Thus we shouldn't try to make it correct for one widget but rely on the common logic. If the size is not correct for browse (assuming the ranges are set correctly) then the thumb size won't be correct for other widgets either and this should be fixed in the common scroll logic.

I had an impression from our discussion that paging works such that you scroll at max position and a subsequent scroll request would trigger load and increase of max scroll. But this doesn't seem to be the case. You only get to max position when you really scroll to the very last record. And this makes sense, the thumb can be used as an indicator whether you hit the end. This is a good news because then we don't have to implement any special logic to allow scrolling beyond max.

From what I've seen on the native 4GL browse widget, step is always the size of one row. That is, a click on scroll button always scrolls one row.

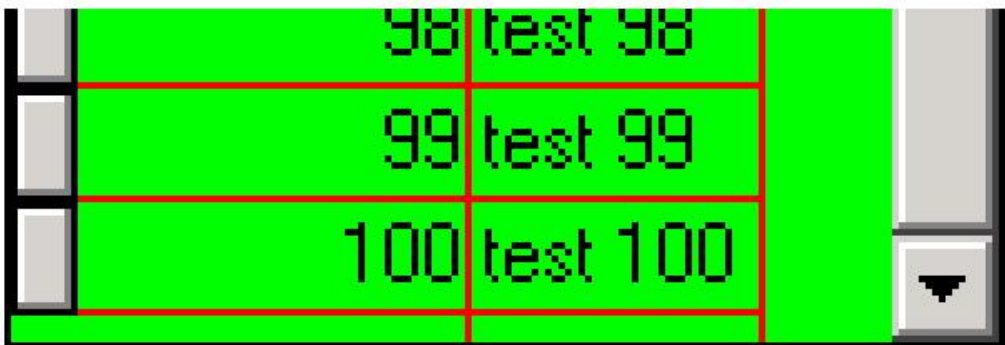
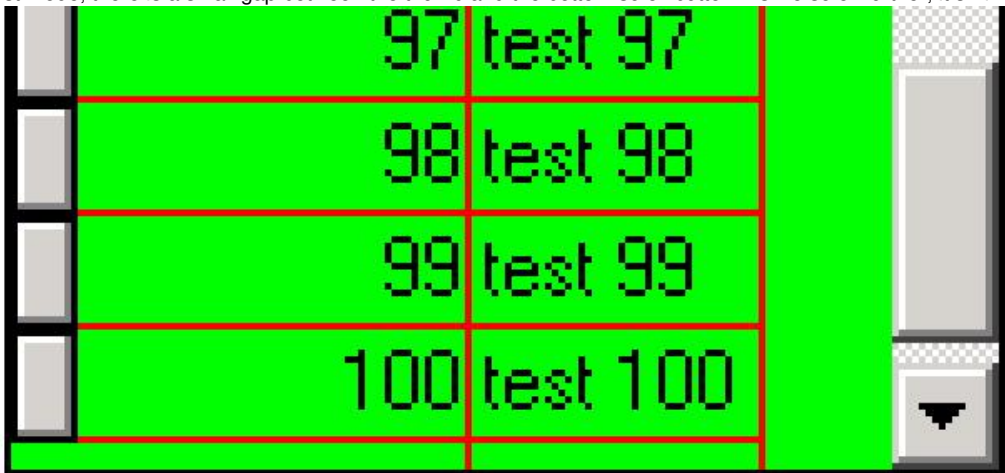
With these assumptions we can utilize the common scroll logic that we already have. Step is given - height of one row, the viewport size is also given, the only variable here is the max value.

I can certainly help you make this work but currently I have to finish more prio task. Meanwhile you can work on something else.

#170 - 10/22/2015 08:08 AM - Stanislav Lomany

- File scrolling-modes.png added

I was wrong. There are actually two scrolling modes: 1. when the number rows is estimated 2. when we exactly know how many rows we have. In the first mode, there is a small gap between the thumb and the bottom scroll button. As we scroll further, it shrinks and disappears at row ~200-300.





**#171 - 10/23/2015 11:05 AM - Stanislav Lomany**

When the number of rows is estimated (rather than we have reached the end), there is the gap at the bottom. Gap area is excluded from Travel size noted in note 166.

The gap size is  $\text{scale}(\text{Travel size} * 0.01)$ . When we pass 100th row, the gap shrinks by  $\text{LOG}_{10}(\text{Top row} + \text{DOWN} - 101) * (\text{Starting gap size} / 3)$  (approximately).

**#172 - 10/23/2015 11:34 AM - Stanislav Lomany**

Greg, here is the issues I'm aware of and you may want pay attention to:

I suppose that this is not directly browse issues:

1. Random crashes "widget is not attached to the top-level window" described in note 90 is still there. We *really* need to fix it.
2. Browse may have some non-drawn areas when is not focused.

Browse issues:

3. One mislayout case.
4. Scrolling with keys when the current row is out of the view - if there are deleted rows, 4GL logic is way too quirky. However we can implement "normal" part.
5. implement 4GL GUI quirk that the first row of a single-selection browse is not selected by default.

Please make directives on these tasks.

Have I missed any other tasks?

**#173 - 10/23/2015 01:34 PM - Greg Shah**

All these items seem to be quite important.

The other thing I would add is to get the scrollbar thumb size/positioning correct as well as to make sure that the scrolling/paging behavior is correct.

**#174 - 10/23/2015 06:20 PM - Hynek Cihlar**

This is a response to note 90.

I can confirm an abend, here is the callstack:

```
Thread [main] (Suspended (exception IllegalStateException))
  ScrollBarGuiButton(AbstractWidget<O>).topLevelWindow() line: 296
  WindowGuiImpl(TopLevelWindow<O>).processEvent(Event) line: 663
  WindowGuiImpl.processEvent(Event) line: 1248
  ThinClient.processProgressEvent(Event) line: 15113
  ThinClient.processEventsWorker() line: 14713
  ThinClient.pop() line: 13745
  ThinClient.eventBracket(boolean, Runnable) line: 13728
  ThinClient.eventDrawingBracket(Widget<?>, boolean, boolean, Runnable) line: 13646
  ThinClient.applyWorker(Event) line: 13402
  ThinClient.waitForEvent(int, boolean, boolean, boolean) line: 12723
  ThinClient.waitForWorker(EventList, int, int, ScreenBuffer, boolean, boolean, boolean, BlockingOperation)
line: 10689
  ThinClient.waitFor(EventList, int, int, ScreenBuffer, BlockingOperation, boolean) line: 10272
  ThinClient.waitFor(EventList, int, int, ScreenBuffer) line: 10226
  NativeMethodAccessorImpl.invoke0(Method, Object, Object[]) line: not available [native method]
  NativeMethodAccessorImpl.invoke(Object, Object[]) line: 62
  DelegatingMethodAccessorImpl.invoke(Object, Object[]) line: 43
  Method.invoke(Object, Object...) line: 497
  MethodInvoker.invoke(Object[]) line: 76
```

```
Dispatcher.processInbound(InboundRequest, boolean, NetResource) line: 705
Conversation.block() line: 319
Conversation.waitMessage(int) line: 257
Queue.transactImpl(Message, int) line: 1128
Queue.transact(Message, int) line: 585
DirectSession(BaseSession).transact(Message, int) line: 223
HighLevelObject.transact(RoutingKey, Object[]) line: 163
RemoteObject$RemoteAccess.invokeCore(Object, Method, Object[]) line: 1425
RemoteObject$RemoteAccess(InvocationStub).invoke(Object, Method, Object[]) line: 97
$Proxy4.standardEntry(ClientParameters) line: not available
ClientCore.start(BootstrapConfig, boolean, boolean) line: 277
```

The problem is that `SwingMouseHandler.mouseMoved()` tries to send `MOUSE_EXITED` event for a widget/top level window already destroyed. `SwingMouseHandler.lastHoveredWidget` must be cleared when such situation occurs.

I will continue with a fix tomorrow.

#### #175 - 10/24/2015 03:44 AM - Hynek Cihlar

Hynek Cihlar wrote:

This is a response to note 90.

I can confirm an abend, here is the callstack:

[...]

The problem is that `SwingMouseHandler.mouseMoved()` tries to send `MOUSE_EXITED` event for a widget/top level window already destroyed. `SwingMouseHandler.lastHoveredWidget` must be cleared when such situation occurs.

Apart from this problem, there is another. We have a race condition there. `SwingMouseHandler.mouseMoved()` (and other implemented `MouseAdapter` methods) is executed on the swing/awt event thread while it unprotectively accesses resources (through the calls to `findMouseSource()`, `processAction()`) together with the main thread.

**#176 - 10/24/2015 03:51 AM - Hynek Cihlar**

Hynek Cihlar wrote:

Hynek Cihlar wrote:

This is a response to note 90.

I can confirm an abend, here is the callstack:

[...]

The problem is that `SwingMouseHandler.mouseMoved()` tries to send `MOUSE_EXITED` event for a widget/top level window already destroyed. `SwingMouseHandler.lastHoveredWidget` must be cleared when such situation occurs.

Apart from this problem, there is another. We have a race condition there. `SwingMouseHandler.mouseMoved()` (and other implemented `MouseAdapter` methods) is executed on the swing/awt event thread while it unprotectively accesses resources (through the calls to `findMouseSource()`, `processAction()`) together with the main thread.

I see two solutions. Synchronize the individual methods, but this is fragile I think. Or introduce new method `EventManager.invokeLater` which would execute the passed in execution block on the event (main) thread in order with the other events.

**#177 - 10/24/2015 04:19 AM - Constantin Asofiei**

Hynek Cihlar wrote:

Hynek Cihlar wrote:

Hynek Cihlar wrote:

This is a response to note 90.

I can confirm an abend, here is the callstack:

[...]

The problem is that `SwingMouseHandler.mouseMoved()` tries to send `MOUSE_EXITED` event for a widget/top level window already destroyed. `SwingMouseHandler.lastHoveredWidget` must be cleared when such situation occurs.

Apart from this problem, there is another. We have a race condition there. `SwingMouseHandler.mouseMoved()` (and other implemented `MouseAdapter` methods) is executed on the swing/awt event thread while it unprotectively accesses resources (through the calls to `findMouseSource()`, `processAction()`) together with the main thread.

I see two solutions. Synchronize the individual methods, but this is fragile I think. Or introduce new method `EventManager.invokeLater` which would execute the passed in execution block on the event (main) thread in order with the other events.

Yes, `EventManager.invokeLater` is the way to go - post a synthetic event with a `Runnable` instance with all the data/code which needs to be executed on the main thread.

**#178 - 10/24/2015 02:22 PM - Hynek Cihlar**

Constantin, should `SwingMouseHandler.processAction()` be called from `SwingMouseHandler.mouse*`() methods when `SwingMouseHandler.canProcessMouse()` returns false?

**#179 - 10/24/2015 02:35 PM - Constantin Asofiei**

Hynek Cihlar wrote:

Constantin, should `SwingMouseHandler.processAction()` be called from `SwingMouseHandler.mouse*`() methods when `SwingMouseHandler.canProcessMouse()` returns false?

I don't think so.. if mouse processing is disabled, then these should be no-op, too. What case of `canProcessMouse()==false` are you referring to,

**#180 - 10/24/2015 02:53 PM - Hynek Cihlar**

Constantin Asofiei wrote:

Hynek Cihlar wrote:

Constantin, should `SwingMouseHandler.processAction()` be called from `SwingMouseHandler.mouse*`() methods when `SwingMouseHandler.canProcessMouse()` returns false?

I don't think so.. if mouse processing is disabled, then these should be no-op, too. What case of `canProcessMouse()==false` are you referring to,

```
public void mouseDragged(MouseEvent e)
{
    if (!processAction(e))
    {
        super.mouseDragged(e);
    }
}
```

super.mouseDragged(e) does check with SwingMouseHandler.canProcessMouse() and if mouse processing is suspended, then it is a no-op. But processAction() is invoked regardless of the potential suspension.

I will fix this as well.

**#181 - 10/24/2015 07:01 PM - Hynek Cihlar**

2677a revision 10959 resolves abend from note 90 and the race condition. Constantin, please review.

**#182 - 10/25/2015 11:23 AM - Constantin Asofiei**

Hynek Cihlar wrote:

2677a revision 10959 resolves abend from note 90 and the race condition. Constantin, please review.

I'm OK with the chanes.

**#183 - 10/27/2015 04:48 AM - Constantin Asofiei**

Stanislav, how difficult is to implement the BROWSE:NUM-ITERATIONS attribute? Is needed for the GUI code.

**#184 - 10/27/2015 05:50 AM - Stanislav Lomany**

I'll make it in a few hours.

**#185 - 10/27/2015 06:06 AM - Stanislav Lomany**

Created task branch 2564d from P2J trunk revision 10950.

**#186 - 10/27/2015 08:21 AM - Stanislav Lomany**

Constantin, BROWSE:NUM-ITERATIONS is implemented in 2564d revision 10951. It comes with layout changes which affect its calculation too.

**#187 - 10/27/2015 10:48 AM - Constantin Asofiei**

Stanislav Lomany wrote:

Constantin, BROWSE:NUM-ITERATIONS is implemented in 2564d revision 10951. It comes with layout changes which affect its calculation too.

Please push them to branch 2677a.

**#188 - 10/27/2015 11:35 AM - Stanislav Lomany**

Merged to branch 2677a as revision 10970.

**#189 - 10/27/2015 05:25 PM - Stanislav Lomany**

Guys, I've checked out 2677a and ran customer's GUI set of tests. The combo box with testcase selection works, but I couldn't find a working testcase (actually I was searching for a testcase with browse). What am I doing wrong?

**#190 - 10/27/2015 05:29 PM - Greg Shah**

Choose item 254. That is the demo scenario we are working on.

**#191 - 10/27/2015 05:53 PM - Stanislav Lomany**

Yes/No question pops up. In either case message closes and nothing happens. There are no exceptions in logs.

**#192 - 10/27/2015 07:01 PM - Greg Shah**

Let's wait until the other issues are cleared enough to get to the first screen with a browse.

Please make a list of the status of the high priority features that are still open and how much work is needed. That work has to happen anyway.

**#193 - 10/27/2015 07:12 PM - Stanislav Lomany**

1. Scrolling with keys when the current row is out of the view ("normal" case) - WIP, tomorrow.
2. implement 4GL GUI quirk that the first row of a single-selection browse is not selected by default - tomorrow.
3. Finish up scrolling - hope tomorrow.
4. Browse may have some non-drawn areas when it is not focused - cannot give estimates.

**#194 - 10/28/2015 02:23 PM - Stanislav Lomany**

Greg, about thumb dragging support: I've noted that normally in 4GL position of a scrollable element is updated as we drag the thumb. However position in a browse is updated only as we released the thumb (is there an option that changes it I'm not aware of?). It is OK to have "normal" dragging for horizontal scrolling. However for vertical scrolling I suppose there are at least two reasons why they did dragging "on release":

1. that saves huge amount of database trips;
2. as we scroll, maximum number of rows can be updated and thumb is supposed to jump from one position to another while it is dragged.

Should I go with dragging "on release"?

**#195 - 10/28/2015 02:27 PM - Constantin Asofiei**

Stanislav, if you are working on scrollbar stuff: at some point the scroll bar buttons are duplicated in the widget tree seen in the Widget Browser (and this means they are duplicated in the scroll container's widget list, too) - the same button (same ID) appears twice in the list. Please take a look and see if you can solve this.

**#196 - 10/28/2015 02:40 PM - Greg Shah**

Should I go with dragging "on release"?

If I understand you, you are saying that is how the 4GL does it for vertical scrolling.

It was not clear to me if the 4GL does this also for horizontal scrolling.

We need to implement browse scrolling exactly like the 4GL does. If they use dragging on release for vertical scrolling, we must do the same. If they

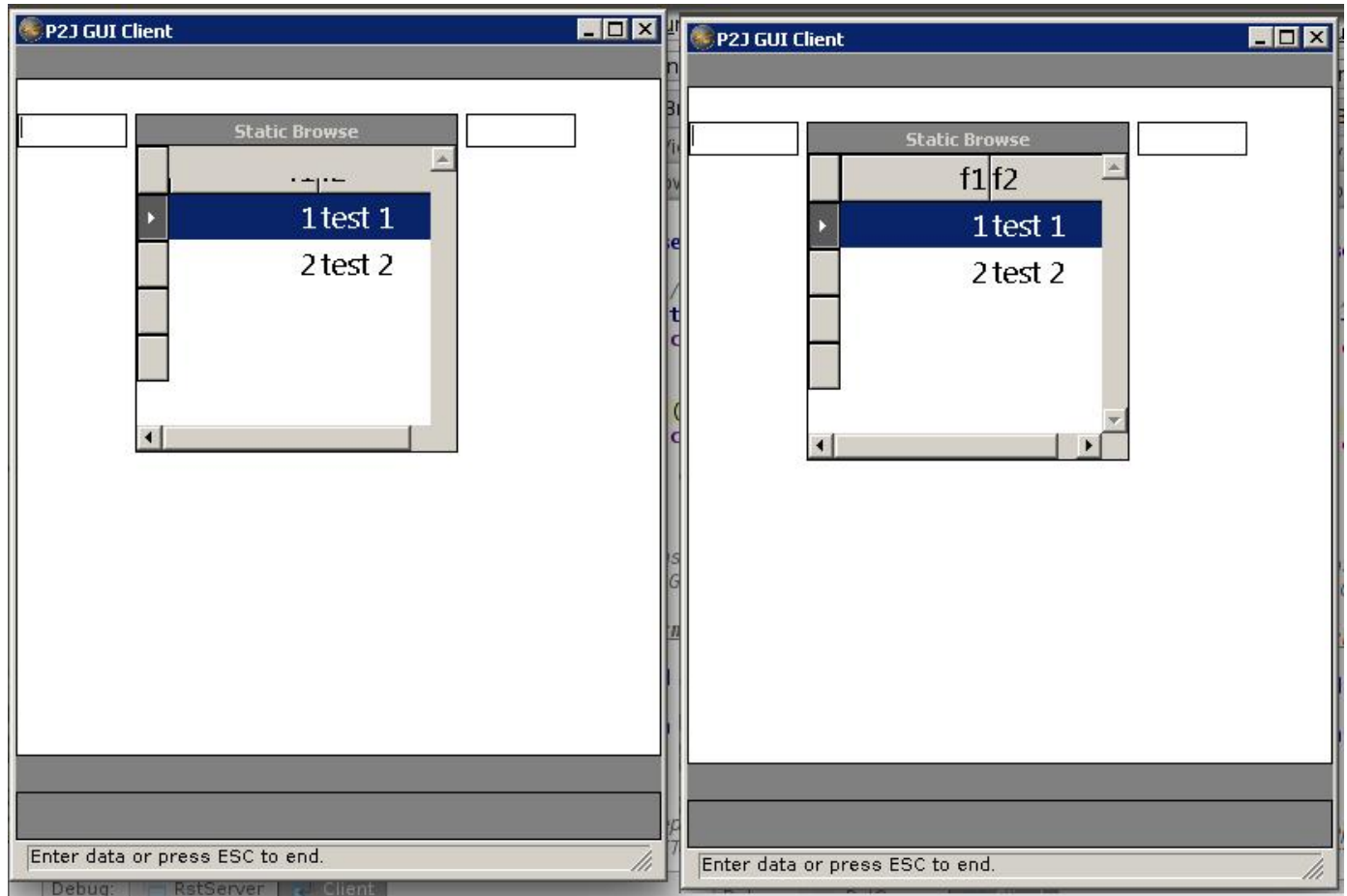
use dragging on release for horizontal scrolling, we must do the same.

#197 - 10/28/2015 03:05 PM - Stanislav Lomany

OK. 4GL uses scrolling "on release" for both scrollbars.

#198 - 10/29/2015 10:25 AM - Stanislav Lomany

- File redraw.png added



Guys, have you met a similar issue or have ideas how to fix it? Non-focused browse is not drawn properly until it is focused or the windows is redrawn.

**#199 - 10/29/2015 04:54 PM - Greg Shah**

As of rev 10954, are issues 1 through 3 (from note 193) complete or is there still work to do on item 3?

**#200 - 10/29/2015 05:25 PM - Stanislav Lomany**

There is work on item 3 to do. Note revision 10954 misses some headers and javadocs.

**#201 - 10/31/2015 05:16 AM - Constantin Asofiei**

Stanislav, another issue: the label can be set on multiple rows via !; i.e. Some!Label!Multiple!Rows will show each word on its own line. This is needed for the BROWSE widget in GUI.

**#202 - 11/02/2015 11:11 AM - Stanislav Lomany**

I'm done with scrolling. Switching to drawing issues and then to multi-line label.

**#203 - 11/02/2015 02:10 PM - Stanislav Lomany**

In ScrollPaneGuiImpl.draw there is the following piece of code which draws scroll bars and THEN performs layout ("draw scroll bars" comment is not quite correct):

```
gd.draw(p, clip, new Runnable()
{
    @Override
    public void run()
    {
        // draw from root
        rootPanel.draw();

        // draw scroll bars
        layoutScrollBars();
    }
}
```

Unless you have objections, I'll swap these lines in order to solve browse drawing problems.

**#204 - 11/02/2015 03:19 PM - Hynek Cihlar**

Stanislav Lomany wrote:

In ScrollPaneGuiImpl.draw there is the following piece of code which draws scroll bars and THEN performs layout ("draw scroll bars" comment is not quite correct):

[...]

Unless you have objections, I'll swap these lines in order to solve browse drawing problems.

Right, layout should be performed before draw. Just be careful about the part that draws the bottom right "intersection rectangle" in layoutScrollBars(),



this should be moved to its own method and performed after rootPanel.draw().

**#205 - 11/02/2015 03:32 PM - Greg Shah**

After the multi-line label issue, please work on the reposition issue in #2272-293.

**#206 - 11/04/2015 06:30 PM - Stanislav Lomany**

Regarding column label: there is the following issue:

```
def temp-table tt field f1 as char column-label "label 1".
```

and

```
def temp-table tt field f1 as char column-label "label 2".
```

in another file is converted to the same temp-table which makes label incorrect for one of the tables. It actually applies to all properties, not only COLUMN-LABEL. IIRC we may already have an open task for that. Should I fix it now?

**#207 - 11/04/2015 06:53 PM - Stanislav Lomany**

Never mind, I found how to fix it.

**#208 - 11/05/2015 12:38 PM - Greg Shah**

Is multi-line label support finished as of branch 2564d rev 10962?

**#209 - 11/05/2015 01:29 PM - Stanislav Lomany**

Yes.

**#210 - 11/05/2015 02:29 PM - Greg Shah**

I'd like to get this merged into 2677a and shift your other bug fixing work to that branch.

Is 2564d ready for code review or does it need some cleanup?

**#211 - 11/05/2015 02:34 PM - Stanislav Lomany**

It is ready. Let me know when to merge it.

**#212 - 11/05/2015 03:16 PM - Greg Shah**

Code Review Task Branch 2564d Revision 10962

I really like the improvements. The ScrollBarController and ScrollBarControllerGuiImpl seem to be a very nice approach to user-controlled customization of the scrolling.

Hynek: please review.

Eric: do you have any concerns with the p2o.xml change? It seems fine to me.

**#213 - 11/05/2015 03:52 PM - Eric Faulhaber**

Greg Shah wrote:

Eric: do you have any concerns with the p2o.xml change? It seems fine to me.

Yes, it's somewhat concerning. If I understand the change, this will generate different implementation classes which are 99.9% the same (identical interface and implementation, only with different annotations), right? This doesn't seem like very good OO.

Why do we need different classes? Is it just for the annotation differences?

I've added Ovidiu as a watcher, since he did the temp-table class hierarchy re-design recently. I'd like his input; unfortunately, he's out until next week.

**#214 - 11/05/2015 04:08 PM - Stanislav Lomany**

My original issue was that the column added by ADD-LIKE-COLUMN had incorrect label.

**#215 - 11/05/2015 04:09 PM - Stanislav Lomany**

The label was specified by COLUMN-LABEL option in temp-table definitions.

**#216 - 11/05/2015 04:28 PM - Eric Faulhaber**

So, is this just about dynamic temp-tables? I'm having trouble understanding how this change (which will change static conversion) addresses this problem. Can you please provide me with a simple 4GL test case or code snippet which illustrates the problem you are trying to solve with this change?

**#217 - 11/05/2015 04:40 PM - Stanislav Lomany**

file1.p:

```
def temp-table tt field f1 as char column-label "label 1".  
...  
browse some-browse:add-like-column("tt.f1").
```

file2.p:

```
def temp-table tt field f1 as char column-label "label 2".  
...
```

Browse in file1.p will have column label "label 2" (depending on conversion order).

**#218 - 11/05/2015 05:02 PM - Eric Faulhaber**

I see. In this case, what does the converted DMO interface and implementation class hierarchy look like for these temp-tables, before and after your fix? Do we have separate interfaces with the fix, or just separate implementation classes?

I believe this change will regress a fix Ovidiu made for shared temp-tables. Please see #2595, starting at note 2. The point of (part of) that issue was to do the opposite of what your change would do; that is, it was meant to ensure that a single DMO definition is created for 2 temp-table definitions that were identical structurally, but differed only by "cosmetic" features, like labels. However, the intention was only to do this for SHARED temp-tables, not for independent tables. It looks like the change left behind the latent bug you have found with independent temp-tables.

Please see if you can come up with a solution that does what you need for independent temp-tables, without regressing the fix described in #2595 for SHARED temp-tables. Unfortunately, I expect it will be more complicated than your current change. It might make sense for you to back out this change for now and defer this bit of work until Ovidiu is back and can contribute to the discussion.

**#219 - 11/05/2015 05:18 PM - Hynek Cihlar**

Greg Shah wrote:

Code Review Task Branch 2564d Revision 10962

I really like the improvements. The ScrollBarController and ScrollBarControllerGuiImpl seem to be a very nice approach to user-controlled customization of the scrolling.

Hynek: please review.

Ad the scrolling changes. This revision introduces a separate alternative implementation of scrolling logic and layout. This is quite unfortunate as we will have to now maintain two different implementations. I don't understand why all this was needed, browse scrolling is no different from scrolling of other widgets.

**#220 - 11/06/2015 05:54 AM - Stanislav Lomany**

I see. In this case, what does the converted DMO interface and implementation class hierarchy look like for these temp-tables, before and after your fix? Do we have separate interfaces with the fix, or just separate implementation classes?

Before:

```
Tt1.java
Tt1_1.java
impl/Tt1_1Impl.hbm.xml
impl/Tt1_1Impl.java
```

After:

```
Tt1.java  
Tt1_1.java  
Tt1_2.java  
impl/Tt1_1Impl.hbm.xml  
impl/Tt1_1Impl.java  
impl/Tt1_2Impl.hbm.xml  
impl/Tt1_2Impl.java
```

I believe this change will regress a fix Ovidiu made for shared temp-tables.

I haven't found an evidence that is true.

It might make sense for you to back out this change for now and defer this bit of work until Ovidiu is back and can contribute to the discussion.

I'll do that.

**#221 - 11/06/2015 06:00 AM - Stanislav Lomany**

Ad the scrolling changes. ... This is quite unfortunate as we will have to now maintain two different implementations.

Actually it might make sense to isolate "standard" panel scrolling logic in a ScrollBarController.

**#222 - 11/06/2015 06:03 AM - Hynek Cihlar**

Stanislav Lomany wrote:

Ad the scrolling changes. ... This is quite unfortunate as we will have to now maintain two different implementations.

Actually it might make sense to isolate "standard" panel scrolling logic in a ScrollBarController.

Yes, I like the idea of ScrollBarController. It is mostly the duplicated code that bothers me.

**#223 - 11/06/2015 06:34 AM - Stanislav Lomany**

There isn't much duplication. Browse scrolling has at least following "features": 1. unconventional thumb size; 2. "gap" area that is used to indicate scrolling towards the end of the result set; 3. drag "on release".

**#224 - 11/06/2015 06:44 AM - Hynek Cihlar**

Stanislav Lomany wrote:

There isn't much duplication. Browse scrolling has at least following "features":

1. unconventional thumb size;

The thumb size is wrong for all widgets, the algorithm for calculating it is common to all. It would have been better to fix it in ScrollBarGuiImpl, we would have had a correct size for all widgets.

2. "gap" area that is used to indicate scrolling towards the end of the result set;

Why is this a special feature? This is just the correct combination of max scroll value and scroll position.

3. drag "on release".

A simpler approach in respect to the original design would have been to have new method ScrollableWidget.scrollFinished() next to ScrollableWidget.scroll().

**#225 - 11/06/2015 06:51 AM - Stanislav Lomany**

I'm not going to argue, I just want to notice that I strongly prefer setting scrollbar parameters directly instead of manipulating them thru panel sizes.

**#226 - 11/06/2015 07:06 AM - Constantin Asofiei**

Stanislav, quick question: is it possible for the BROWSE in GUI to display a partial row on top/bottom edge?

**#227 - 11/06/2015 07:10 AM - Stanislav Lomany**

On top - no. On bottom - 1 px can be hidden.

**#228 - 11/06/2015 07:22 AM - Constantin Asofiei**

Stanislav Lomany wrote:

On top - no. On bottom - 1 px can be hidden.

OK, then to me this seems a different case of scrolling than the case of frame content scroll. In frame's case, the entire frame body needs to be drawn and the scrollbars manipulate the viewport's location so that it matches the visible area.

In EDITOR case (and for BROWSE I assume, too) only the visible content is drawn, and some internal indexes are kept so that this is in sync with the current scroll position.

**#229 - 11/06/2015 09:00 AM - Greg Shah**

OK, I think the current approach seems the reasonable thing to do right now. Perhaps there is a way to reduce the duplicated code, but we can address that later. There do seem to be some requirements for BROWSE (and EDITOR) that may be different from the standard scrolling that is being done for other widgets. We need some time to consider these requirements in the scope of that de-duplication effort. I'm going to create a task for this, so we don't forget to look at it later.

**#230 - 11/06/2015 09:10 AM - Greg Shah**

Considering the intrusiveness of the scrolling changes (including non-GUI code changes), I think it makes sense to get 2564d runtime regression tested before it is merged into 2677a.

Stanislav: What do you think?

**#231 - 11/06/2015 09:12 AM - Stanislav Lomany**

Agree. I'll run regression testing.

**#232 - 11/11/2015 04:22 PM - Ovidiu Maxiniuc**

Related to Eric & Stanislav discussion on notes 214-220:

I created some testcases based on Stanislav's hints.

The conversion was not fine for temp-tables. Because the column-label is not taken into consideration when differentiating interfaces only a single DMO is declared and the additional information about the column label of the second table is lost.

I introduced the init attribute in one of the files to force the algorithm create different DMOs. Indeed, they share the same interface and have different Impl@s. This is also reflected in the @LegacyField/columnLabel annotation for the DMOs.

```
@LegacyField(fieldId = 1, name = "f0", columnLabel = "label 0")
&
@LegacyField(fieldId = 1, name = "f0", initial = "1001", columnLabel = "abc")
```

In each (external) procedure, the temp-table is instantiated locally:

```
Tt1_1.Buf tt = TemporaryBuffer.define(Tt1_1.Buf.class, "tt", "tt", false);
&
Tt1_2.Buf tt = TemporaryBuffer.define(Tt1_2.Buf.class, "tt", "tt", false);
```

In this case the correct label is possible to be displayed.

I was not aware that this kind of methods exists. I assumed that all 'presentation' information is available at conversion time directly from 'local' AST tree. However, in this case the character argument of the ADD-LIKE-COLUMN method forces a dynamic update of the browse, at runtime.

Don't know which is the good solution here. I tried to groups as much as possible the code but we are already duplicating much of the DMO code in the Impl classes. Adding the 'presentation' attributes as a criteria will create more classes, only for the character annotation. I wonder if a xml can be used to store the missing information at runtime.

#### **#233 - 11/11/2015 04:44 PM - Eric Faulhaber**

Yes, it seems the fundamental problem is that when we introduced Java annotations to the DMO implementation classes, we assumed a 1-to-1 association between converted field and each attribute. In retrospect, even though this may be how it is implemented in Progress, implementing based on this assumption probably was a bad idea.

We have a nice inheritance model in Java that allows us to define classes based on their data structures. I really want to avoid creating multiple classes which are structurally identical but differ only in their annotations. Usually adding a level of abstraction is the solution for problems like this, but I'm reluctant to complicate the 99.9% of DMOs that work well with the current model in order to deal with this edge case, which seems limited to non-shared temp-tables.

I don't have an answer yet; I need some time to think about this. Any ideas are welcome...

#### **#234 - 12/01/2015 01:34 PM - Stanislav Lomany**

- Status changed from New to WIP

Remaining issues.

Will make soon:

1. #2825 fix is ready and tested, can be merged to trunk after 2677b
2. #2882 WIDTH-CHARS for browse column

Following issue should be done in sequence:

3. Browse editing
4. #2630 save rows in browse
5. #2799 browse validation support

This can go at any time after item 3.

6. #2628 non-fillin support in browse columns.

These are minor issues that can be addressed independently:

7. #2896 buffer scoping issue remains
8. #2892 Scrolling with mouse wheel.
9. #2893 Mouse clicks on rows are not 100% accurate.
10. #2894 Multiple selection with Shift and Ctrl keys.

Do not touch until we meet it:

11. #2895 COLUMN-LABEL annotation for dynamic queries (notes 217-233 above).

It's hard to say what is more prioritized in the set 7, 8, 9, 10, [3-5], 6. It probably depends on what is met more often, or goes first in the customer's applications: editing, multi-selection or non-fillin browse. So please prioritize it.

**#235 - 12/01/2015 02:37 PM - Greg Shah**

The priority of 1 through 6 is correct. I have no specific opinion on 7 through 11. We can leave it as is for now.

Please create separate tasks for 8 through 11, so they can be tracked individually. Make them related to this task and add me as a watcher.

**#236 - 12/01/2015 02:37 PM - Greg Shah**

Also, create a new sub-task of #2811 for the buffer scoping issue. I want to close #2819.

**#237 - 12/07/2015 02:06 PM - Stanislav Lomany**

Created task branch 2564e from trunk revision 10957. This branch will contain implementation of editing browse.

**#238 - 12/08/2015 02:04 PM - Stanislav Lomany**

- File editing1.png added

Positioning of the fill-in in a cell:



Single Browse		
	f1	f2
▶	1	test 1

Single Browse		
	f1	f2
▶	1	test 1

#239 - 12/08/2015 02:05 PM - Stanislav Lomany

- File deleted (editing1.png)

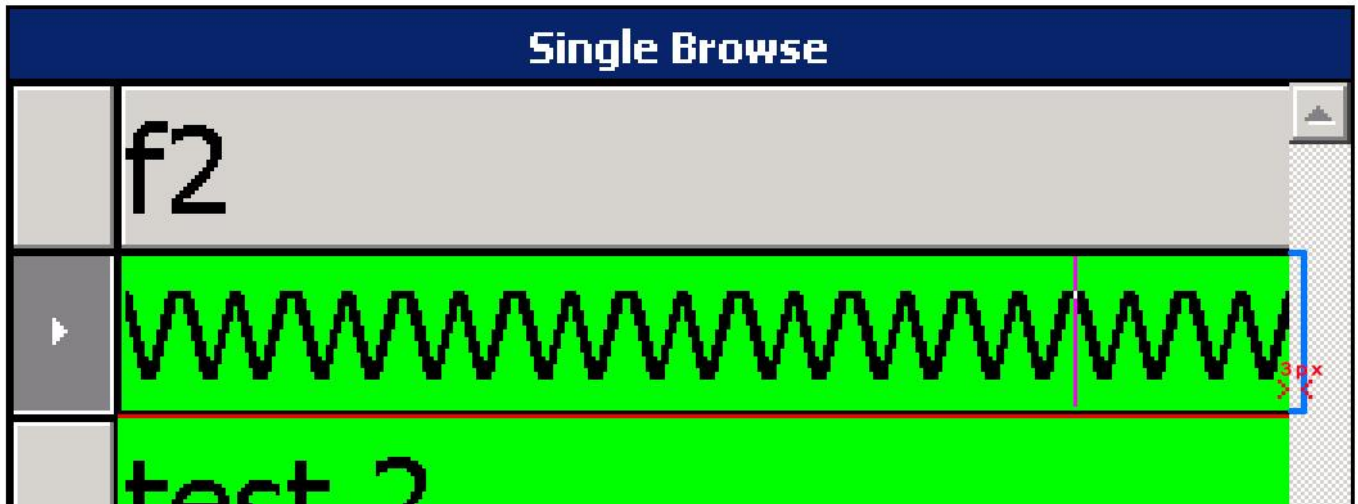
#240 - 12/08/2015 02:07 PM - Stanislav Lomany

- File editing1.png added

#241 - 12/08/2015 03:40 PM - Stanislav Lomany

- File editing2.png added

If a column doesn't fit into visible area, fill-in can be extended to invisible area to the right, maximum 3 pixels (not including fill-in border).



#242 - 12/15/2015 06:24 PM - Stanislav Lomany

- File edit-quirk.png added

Guys, there is the quirk: in COLUMN-SCROLLING mode, if a cell is edited and we scroll to the right, the cell is shrunk so that the edited text is left-aligned to the viewport edge. If we press a button, viewport is scrolled to fully display the column (this part is a common behavior). Should I implement "cell shrinking quirk" now or leave for later?

Single Browse	
f1	f2
TEST	test 1
test 2	test 2

Single Browse	
	f2
TEST	test 1
t 2	test 2

Single Browse	
f1	f2
TESTX	test 1
test 2	test 2

#243 - 12/17/2015 11:12 AM - Greg Shah

Should I implement "cell shrinking quirk" now or leave for later?

This seems to be a pretty strange behavior that most users would not like. Please create a quirk task for this to handle it later.

**#244 - 12/17/2015 12:49 PM - Stanislav Lomany**

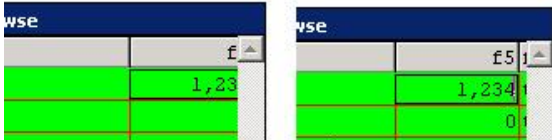
The issue was moved to [#2936](#).

**#245 - 12/18/2015 02:18 PM - Stanislav Lomany**

- File *edit-right-trim.png* added

"Scroll to view" rules:

1. If the current (editable) column is not "visible" in the browse viewport and user *a*) pressed a key or *b*) switched to another cell in this column (using mouse or keyboard), the viewport is scrolled to the this target column so it becomes
  - a) **left-aligned** in the viewport if the column is to the left of the view or
  - b) **right-aligned** if it is NO-COLUMN-SCROLLING mode and the column is to the right of the view or
  - c) **rightmost of the full columns** if it is COLUMN-SCROLLING mode and the column is to the right of the view.
2. Column is not "visible" if any part on its left side is hidden or more than  $\sim 1.9 * \text{browse\_font\_width}$  pixels is hidden from its right part. In this example you can type text in any of these columns and they are not scrolled to view:



3. Column text alignment doesn't affect scrolling.
4. If a locked column is edited (i.e. user presses a key or changes cell), position of scrolling columns is reset.

**#246 - 12/23/2015 05:42 AM - Stanislav Lomany**

Guys, consider a case when a cell is edited and we have scrolled so this cell is outside of the view (fully or partially). In this case, when we press a key, the browse should be scrolled to show the cell in the view. It is scrolled as soon as start holding down a key, including CTRL and SHIFT. However `EventManager.getEvent()` doesn't seem to support "button down" events. What should I do at this point?

**#247 - 12/26/2015 04:39 PM - Stanislav Lomany**

Side note: there are some issues which are not browse issues but affect browse. Most of them are fill-in issues: cursor positioning, selection, zap mode etc. There are many of them, so it doesn't even make sense to record them. And there is another common issue: GUI 4GL displays validation messages as alert boxes rather than messages in the message area. Example:

```
def temp-table tt field f1 as INTEGER FORMAT ">, >>>9"
    index idx1 is unique f1.

CREATE tt. tt.f1 = 1.
CREATE tt.

MESSAGE "press 1, ENTER".
UPDATE tt.f1 WITH FRAME f1.
```

**#248 - 12/29/2015 07:02 PM - Stanislav Lomany**

Current known problems with browse editing:

1. Scroll to the edited row after a key is down (see note 246).
2. Proper scroll to the edited row which is below the current view.
3. Proper implementation of Page Up and Page Down for editing browse.
4. Bug with endless scroll for a compound adaptive query.
5. Check row deletion.
6. Implement INSERT-ROW.

**#249 - 12/31/2015 07:48 AM - Stanislav Lomany**

Rebased task branch 2564e from P2J trunk revision 10961.

**#250 - 12/31/2015 09:05 AM - Stanislav Lomany**

Please review branch 2564e revision 10973.

**#251 - 01/05/2016 07:20 AM - Greg Shah**

And there is another common issue: GUI 4GL displays validation messages as alert boxes rather than messages in the message area.

I think you are just seeing an artifact of the procedure editor. Normal runtime support does not work that way. Did you run the procedure directly from the command line using the -p switch?

**#252 - 01/05/2016 08:37 AM - Greg Shah**

Code Review Task Branch 2564e Revision 10973

The changes are quite good. It may make sense to go through regression testing and get this into the trunk (with the changes requested below). What do you think?

1. The history entry for ThinClient states that the change is related to editable browse, but the change in ThinClient.findLegacySource() seems like it would affect a non-editing browse that happens to be in a frame that has other enabled widgets. The Browse.getFocusedWidget() does seem to return a fillin or the browse itself depending on the returned value from getActiveEditor(). So I guess this is OK. My primary concern is that the ThinClient.findLegacySource() probably needs a comment to explain that the returned widget can be a fillin or browse (and the condition upon which the returned value depends).

2. Please put curly braces around the code at Browse.java line 2633. It looks a bit too "python" for me. :)

```
else
    editor.setState(ScreenBuffer.UNINITIALIZED);
editor.setUnderline(false);
```

3. Please replace the newly added references to Java2D and Graphics2D in BrowseColumnGuiImpl. The color modifications are not specific to Swing right? As far as I can tell, your changes should work properly in the web client too, so I don't want readers confused (thinking that the code is Swing-specific).

4. I suspect that BrowseGuiImpl.displayErrorMessage() should not be hard coded to ThinClient.messageBox() (as discussed in note 251).

**#253 - 01/06/2016 05:43 AM - Stanislav Lomany**

The changes are quite good. It may make sense to go through regression testing and get this into the trunk (with the changes requested below). What do you think?

This is a good idea. I'll run regression testing.  
BTW editing browse doesn't work properly in web version.

3. Please replace the newly added references to Java2D and Graphics2D in BrowseColumnGuiImpl. The color modifications are not specific to Swing right? As far as I can tell, your changes should work properly in the web client too, so I don't want readers confused (thinking that the code is Swing-specific).

Color modifications are properly reproduced by web version. I've corrected the comment.

4. I suspect that BrowseGuiImpl.displayErrorMessage() should not be hard coded to ThinClient.messageBox() (as discussed in note 251).

Yes, you are right, that it is an editor quirk.

**#254 - 01/06/2016 09:24 AM - Greg Shah**

Code Review Task Branch 2564e Revision 10974

The changes are good. I think it is best to just remove BrowseGuiImpl.displayErrorMessage().

**#255 - 01/06/2016 09:39 AM - Constantin Asofie**

Stanislav Lomany wrote:

Guys, consider a case when a cell is edited and we have scrolled so this cell is outside of the view (fully or partially). In this case, when we press a key, the browse should be scrolled to show the cell in the view. It is scrolled as soon as start holding down a key, including CTRL and SHIFT. However EventManager.getEvent() doesn't seem to support "button down" events.

In regard to key pressed support in P2J: currently there are two type of events, EventType.KEY\_PRESSED and EventType.KEY\_TYPED. When a

physical key is pressed, `EventManager.getKeyEvent` gets called which sets the `KEY_PRESSED` for any key code with `key > 255 || key < ' '`, otherwise it sets it to `KEY_TYPED`. So, we don't have "key pressed" support yet in P2J, as the `SwingKeyboardReader` doesn't expose "key press" events to the P2J runtime. Beside this, `KEY_PRESSED` is hard-coded in misc other places, as an argument for the `KeyInput` c'tor (but I don't think this hurts). Notification for these events are sent to `KeyListener` implementations, via `AbstractWidget.processKeyListeners` - so, if (for example) an editing cell in a `BROWSE` has a `KeyListener` registered, it will be able to distinguish between key press and key typed, once we change our keyboard impl to send these kind of events, too.

But this does not easily help for modifier keys (`CTRL`/`SHIFT`/`ALT`) - these are not emitted by the `SwingKeyboardReader` implementation at all (and if at some point we decide to emit them, we need to be very careful to not include them in `ANY-KEY` triggers, and maybe other cases, too).

**#256 - 01/07/2016 04:14 AM - Stanislav Lomany**

The changes are good. I think it is best to just remove `BrowseGuiImpl.displayErrorMessage()`.

Done. The regression testing passed. Should I check in?

**#257 - 01/07/2016 05:18 AM - Stanislav Lomany**

Greg, please tell me if I should implement `CTRL` and `SHIFT` (not `ALT`) `KEY_PRESSED` events. The use case of it is when a user presses some keyboard shortcut containing these keys (like `CTRL+TAB`), `browse` is naturally scrolled to the edited row.

**#258 - 01/07/2016 09:17 AM - Greg Shah**

Stanislav Lomany wrote:

The changes are good. I think it is best to just remove `BrowseGuiImpl.displayErrorMessage()`.

Done. The regression testing passed. Should I check in?

Have you confirmed that the changes don't cause any problems with the 454 testcase? Please check that and if it is OK, then you can merge to trunk.

**#259 - 01/07/2016 09:19 AM - Greg Shah**

Stanislav Lomany wrote:

Greg, please tell me if I should implement CTRL and SHIFT (not ALT) KEY\_PRESSED events. The use case of it is when a user presses some keyboard shortcut containing these keys (like CTRL+TAB), browse is naturally scrolled to the edited row.

We may need to implement this eventually, but I think we can wait for now. Please create a new task for it. Link it here and to the quirks task as a parent. I'm not sure it is really a "quirk", but this way we will have a place to find it again.

#### **#260 - 01/07/2016 10:03 AM - Stanislav Lomany**

Task branch 2564e committed into the trunk as bzd revision 10962.

#### **#261 - 01/07/2016 10:45 AM - Greg Shah**

Fixing browse editing mode in the web client should be your top priority.

After that, you can resolve the other issues with editing in note 248.

Then, I guess the issues 4 through 10 from note 234 would follow.

#### **#262 - 01/10/2016 04:07 PM - Stanislav Lomany**

Created task branch 2564f from P2J trunk revision 10962.

#### **#263 - 01/11/2016 05:32 PM - Stanislav Lomany**

Sometimes when a row is saved in a browse backed by an adaptive query, then, when we continue to scroll browse, data set becomes incorrect. Here is the testcase which reproduces the problem:

```
def temp-table tt field f1 as INTEGER
                index idx1 is unique f1.
```

```
DEF VAR i AS INTEGER.
REPEAT i = 1 TO 20:
  create tt. tt.f1 = i.
END.
```

```
DEF QUERY q FOR tt scrolling.
OPEN QUERY q FOR EACH tt.
```

```
get next q.
get next q.
get next q.
get next q.
```

```
reposition q to row 3.
get next q.
tt.f1 = 333.
```

```
reposition q to row 1.
get next q.
```

```
reposition q to row 5.
get next q.
message string(tt.f1).
get next q.
```



```
message string(tt.f1).
get next q.
message string(tt.f1).
```

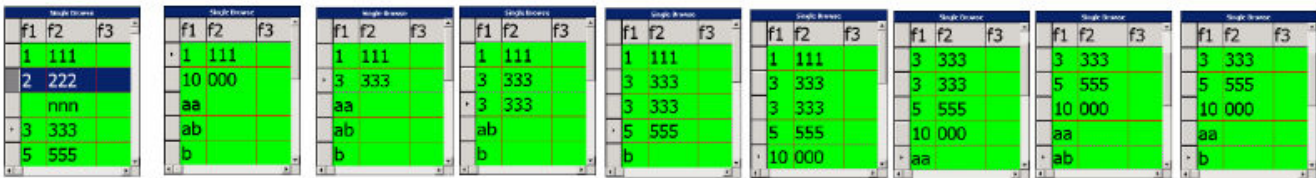
P2J output: 2 4 5  
 4GL output: 5 6 7

Working on that.

**#264 - 01/21/2016 06:19 AM - Stanislav Lomany**

- File deletion.p added
- File deletion.jpg added

FYI, when it comes to deletion of a row when a new row presents in the result set, consequences may be unpredictable. E.g. in the attached testcase pressing "a" to delete the selected row on the first screen makes the state of the browse inconsistent. The following screens demonstrate "content" of the browse after this operation as we scroll it further (one screen per one row down).



**#265 - 01/21/2016 11:35 AM - Greg Shah**

Stanislav: Would you please summarize the status of the options/attributes/methods listed in note 1? For each one, please note the level of support (none/conversion/runtime/all) as well as whether any of the support is partial. In addition, I need to know if any of that support is in 2564f (as opposed to the trunk).

**#266 - 01/21/2016 01:00 PM - Guy M**

Couple of things related to #264

Firstly, how was the new line added? Is there some missing code from the example procedure? The only way I could add a new line was by adding some code e.g.

ON 'b' anywhere

DO:

BROWSE brws:INSERT-ROW().

END.

Secondly, when I did that, I didn't see the same results, although the newly added line "disappeared" along with the current row.

I searched the Progress KB for INSERT-ROW and I found the following: [\[\[http://knowledgebase.progress.com/articles/Article/21465\]\]](http://knowledgebase.progress.com/articles/Article/21465). This seems to indicate that if you use INSERT-ROW, you also need to add code to write a new record back to the DB. So you need to both create the new record and call CREATE-RESULT-LIST-ENTRY. I guess if you do the same, your browse will behave better.

That all said, I don't think we use updateable browses within our product (or if so, very rarely) - I think because we found issues with them years ago. There probably are some good candidates for using updateable browses (e.g. data entry screens), but I don't think our BPM product provides an updateable grid, so it would lead to inconsistencies and possibly a retrograde appearance when we finally convert to BPM.

**#267 - 01/21/2016 01:21 PM - Greg Shah**

OK, then we will treat this as another "quirk".

Stanislav: please add a sub-task to the quirks task, documenting this issue.

**#268 - 01/22/2016 08:38 AM - Stanislav Lomany**

Stanislav: Would you please summarize the status of the options/attributes/methods listed in note 1

Options

FIT-LAST-COLUMN - Full support  
NO-COLUMN-SCROLLING - Full support  
ROW-HEIGHT-CHARS - Full support, however one graphic artefact needs to be fixed  
COLUMN-BGCOLOR - I think the option is named BGCOLOR - Full support  
EXPANDABLE - Full support  
NO-SCROLLBAR-VERTICAL - No support. Does in present in the customer's systems?

Methods

MOVE-COLUMN - Conversion support.

**#269 - 01/22/2016 08:55 AM - Greg Shah**

COLUMN-BGCOLOR - I think the option is named BGCOLOR - Full support

There is actually a real option in use in the customer app that is named COLUMN-BGCOLOR. Please make sure we are handling it.

NO-SCROLLBAR-VERTICAL - No support. Does in present in ...?

No, it has been removed in recent cut-down work.

**#270 - 01/22/2016 08:58 AM - Greg Shah**

Please note that the SCROLLBAR-VERTICAL attribute is used on browse in the customer app. Since this needs to be supported anyway, do add support for the NO-SCROLLBAR-VERTICAL option as well.

**#271 - 01/22/2016 09:41 AM - Stanislav Lomany**

There is actually a real option in use in the customer app that is named COLUMN-BGCOLOR. Please make sure we are handling it.

OK, it is supported actually.

**#272 - 01/22/2016 05:52 PM - Stanislav Lomany**

Attributes

ROW-HEIGHT-CHARS - Supported until browse is realized. Do we want to implement the ability to update this attribute after the browse has been realized?  
ALLOW-COLUMN-SEARCHING - Conversion support.  
SEPARATORS - Full support.  
EXPANDABLE - Full support.  
COLUMN-RESIZABLE - Conversion support.  
LABEL-BGCOLOR - Full support.  
COLUMN-MOVABLE - Conversion support.  
FIT-LAST-COLUMN - Full support.  
MAX-CHARS - Conversion support.  
SCROLLBAR-VERTICAL - Conversion support  
COLUMN-FGCOLOR - No support. Well, there is the actual support of this type of coloring, but it is not linked to this attribute.  
NUM-ITERATIONS - Full support.  
DOWN - Conversion support

**#273 - 01/25/2016 09:31 AM - Greg Shah**

It is my understanding that the row deletion support is complete. This means that the remaining critical browse editing support is [#2955](#) and [#2630](#). Correct?

Those changes should be added to 2564f.

The next highest priority is providing full conversion and runtime support for the options/attributes/methods listed in this task.

At that point 2564f needs to be regression tested and merged to trunk. We will close [#2564](#) when that is done.

What is your estimate for the time needed for this completion? It is a bit of a surprise to me that these options/attributes/methods were still not complete, so I had not planned for the extra time needed there.

#274 - 01/25/2016 09:57 AM - Stanislav Lomany

It is my understanding that the row deletion support is complete. This means that the remaining critical browse editing support is [#2955](#) and [#2630](#). Correct?

Yes.

What is your estimate for the time needed for this completion? It is a bit of a surprise to me that these options/attributes/methods were still not complete, so I had not planned for the extra time needed there.

A bit more than a week.

About attributes:

ALLOW-COLUMN-SEARCHING - Requires search ability (whatever this means).

COLUMN-RESIZABLE - Requires ability to resize columns with mouse.

COLUMN-MOVABLE - Requires ability to move columns with mouse.

MAX-CHARS - Requires ability to have drop-down columns.

So I'm unable to implement these attributes so far.

MOVE-COLUMN - that function may take up to 2 days.

Other attributes is not a problem and can be done quickly.

And one more issue which I found recently: if **current** unsaved row goes out of the view, we can scroll back to it (other unsaved rows disappear). Should I implement this ability?

#275 - 01/25/2016 11:26 AM - Greg Shah

ALLOW-COLUMN-SEARCHING - Requires search ability (whatever this means).

Check if this is the same as [#2959](#).

COLUMN-RESIZABLE - Requires ability to resize columns with mouse.

COLUMN-MOVABLE - Requires ability to move columns with mouse.

This is just something to add to browse, right? It isn't dependent on anything else, so it should be implemented now.

MAX-CHARS - Requires ability to have drop-down columns.

OK, make a note in [#2628](#) that the MAX-CHARS runtime support must be added in that task.

**#276 - 01/25/2016 12:00 PM - Stanislav Lomany**

Check if this is the same as [#2959](#).

The behavior described in this task is *disabled* if ALLOW-COLUMN-SEARCHING is ON. Documentation says that

If ALLOW-COLUMN-SEARCHING is set to TRUE, the START-SEARCH and END-SEARCH events will be triggered when a search is initiated and completed.

so I'll check how events are triggered/applied when I'll start this task.

**#277 - 01/25/2016 12:41 PM - Greg Shah**

It seems (from the following article) that the attribute only causes the events to be generated in response to a mouse click on the column header.

<http://knowledgebase.progress.com/articles/Article/P73465>

Notice how the user's triggers must provide all the real functionality to close the old query and associate a new open query with different sorting. It just seems to integrate the event processing (and maybe some repainting?). I guess it is needed because the user cannot otherwise hook a click on the browse column header alone.

**#278 - 03/04/2016 02:33 PM - Stanislav Lomany**

Rebased task branch 2564f from P2J trunk revision 10988.

**#279 - 03/23/2016 01:48 PM - Greg Shah**

Does 2564f have full support for COLUMN-MOVABLE, COLUMN-RESIZABLE, CREATE-ON-ADD, MIN-HEIGHT-CHARS and MOVE-COLUMN?

**#280 - 03/23/2016 01:54 PM - Greg Shah**

Also, does 2564f support the NO-SCROLLBAR-VERTICAL option?

**#281 - 03/23/2016 02:24 PM - Stanislav Lomany**

No, this options are not supported in 2564f.

**#282 - 04/07/2016 10:19 AM - Stanislav Lomany**

Rebased task branch 2564f from P2J trunk revision 10996.

**#283 - 04/07/2016 12:39 PM - Greg Shah**

Code Review Task Branch 2564f Revision 11016

I'm good with the UI changes. Eric will review the persistence stuff.

One minor issue: BrowseColumnGuiImpl needs a history entry.

**#284 - 04/07/2016 02:23 PM - Eric Faulhaber**

Code review 2564f/11016 (persistence):

The updates look clean, but there are some pretty fundamental functional changes being made throughout all the major query types, and I simply can't tell the impact from reading the code. This will have to be tested thoroughly.

What is/are the use case(s) for the ParameterFilter?

Please help me understand the implications of the changes to the use of ChangeBroker in CompoundQuery. Specifically, I'm confused by this during cleanup:

```
ChangeBroker.get().removeFromGlobal(this);
```

Where was this added to the global ChangeBroker listener scope? Any changes to the use of the ChangeBroker have the potential to cause memory leaks, so I want to understand this better.

The changes to CompoundQuery don't seem to be limited to browse cases. Did you find this behavior was needed in all cases? Or is this only for a certain type of CompoundQuery? If so, I don't see the conditional logic related to this.

Likewise, is the change to Cursor.forward a general-purpose fix, or does it need to be narrowed in scope?

What is the use case for FieldReference.getTemp?

Thanks.

The changes to CompoundQuery don't seem to be limited to browse cases. Did you find this behavior was needed in all cases? Or is this only for a certain type of CompoundQuery? If so, I don't see the conditional logic related to this.

Likewise, is the change to Cursor.forward a general-purpose fix, or does it need to be narrowed in scope?

This is a general-purpose update for non-preselect queries that addresses problems with subsequent query iterations when the current record of a participating buffer was updated.

Browse isn't directly related to, but it was easy for me to expose this issue with an editable browse.

Testcases can be executed using

```
uast/adaptive_scrolling/adaptive_use_same_buffer/adaptive-scroll-harness.p  
uast/adaptive_scrolling/adaptive_use_same_buffer/adaptive-noscroll-harness.p
```

What is/are the use case(s) for the ParameterFilter?

That is an object that can be passed to a sub-query and return actual or snapshot values of the parameters for that sub-query basing on the state of the parent query. See CompoundQuery.SnapshotParameterFilter.

Please help me understand the implications of the changes to the use of ChangeBroker in CompoundQuery. Specifically, I'm confused by this during cleanup:

[...]

Where was this added to the global ChangeBroker listener scope? Any changes to the use of the ChangeBroker have the potential to cause memory leaks, so I want to understand this better.

That is cleanup for

```
ChangeBroker.get().addListener(this);  
if (blockDepth == 0)  
{  
    unregisterOnCleanup = true;  
}
```

in CompoundQuery.initReferenceRowSupport.

What is the use case for FieldReference.getTemp?

Get the value of the specific field of a snapshot. Unfortunately FieldReference.get(DataModelObject) also sets this DMO as the future referent.

**#286 - 04/07/2016 03:35 PM - Greg Shah**

While you are discussing these issues with Eric, please go ahead and put this branch into regression testing to save time later.

**#287 - 04/07/2016 04:04 PM - Greg Shah**

Eric is away from his desk, but he wants me to relay that he is OK with your answers in note 285.

He will test your code using the customer's app later tonight.

**#288 - 04/07/2016 04:21 PM - Stanislav Lomany**

Rebased task branch 2564f from P2J trunk revision 10997. Headers are fixed.

**#289 - 04/07/2016 07:10 PM - Stanislav Lomany**

After rebase there are at least 3 browse GUI regressions/issues.

**#290 - 04/07/2016 10:29 PM - Eric Faulhaber**

Stanislav Lomany wrote:

After rebase there are at least 3 browse GUI regressions/issues.

I'll hold off regression testing with the customer's server app until you have a final candidate.

Are any of the regressions related to persistence, or are they all visual?

**#291 - 04/08/2016 04:25 AM - Stanislav Lomany**

All visual.

**#292 - 04/08/2016 06:40 PM - Stanislav Lomany**

FYI, 2564f has passed regression testing.

**#293 - 04/08/2016 07:23 PM - Greg Shah**

Great! Are you close on the remaining two GUI regressions?

**#294 - 04/08/2016 07:24 PM - Stanislav Lomany**

I'm not yet sure, but I'll continue with it tomorrow.

**#295 - 04/08/2016 07:32 PM - Greg Shah**

OK, how about this: please merge 2564f to trunk now. This will let Hynek rebase and finish work on 2954a tomorrow.

Please document the remaining GUI regressions in [#3038](#) and include the fixes there (in branch 3038a). This way we can close 2564.

Do you have any concerns with that plan?



**#296 - 04/08/2016 07:46 PM - Stanislav Lomany**

Sounds good!

**#297 - 04/08/2016 07:53 PM - Greg Shah**

Cool. Please merge 2564f to trunk.

**#298 - 04/08/2016 08:38 PM - Stanislav Lomany**

Rebased task branch 2564f from P2J trunk revision 10998.

**#299 - 04/08/2016 08:49 PM - Stanislav Lomany**

Task branch 2564f committed into the trunk as bzt revision 10999.

**#300 - 04/09/2016 03:19 AM - Greg Shah**

- Status changed from WIP to Closed

**#301 - 11/16/2016 12:12 PM - Greg Shah**

- Target version changed from Milestone 12 to GUI Support for a Complex ADM2 App

**Files**

browse.png	4.24 KB	08/31/2015	Stanislav Lomany
browse-metricis.png	7.24 KB	09/07/2015	Stanislav Lomany
browse1.png	7.18 KB	09/09/2015	Stanislav Lomany
browse2.png	17.3 KB	09/09/2015	Stanislav Lomany
browse-edit1.png	9.79 KB	09/10/2015	Stanislav Lomany
compare.png	5.87 KB	09/11/2015	Stanislav Lomany
fillin-diff.png	16.1 KB	09/23/2015	Stanislav Lomany
browse-cell-offset.ods	39.1 KB	09/23/2015	Stanislav Lomany
double-draw.png	17.1 KB	09/27/2015	Stanislav Lomany
w.png	26.8 KB	09/28/2015	Stanislav Lomany
selection-color.png	3.65 KB	09/29/2015	Stanislav Lomany
scroll-error.png	3.26 KB	09/30/2015	Stanislav Lomany
browse-gui-stat1.p	1.26 KB	09/30/2015	Stanislav Lomany
BrowseGuiImpl.java.diff	6.92 KB	10/13/2015	Hynek Cihlar
brws-size-implicit-dyn.p	1.17 KB	10/14/2015	Stanislav Lomany
go.p	664 Bytes	10/14/2015	Stanislav Lomany
brws-size-chars.png	8.76 KB	10/21/2015	Stanislav Lomany
browse_scroll_20_db_entries_5_height.png	8.32 KB	10/21/2015	Hynek Cihlar
browse_scroll_8_db_entries_15_height.png	12.2 KB	10/21/2015	Hynek Cihlar
scrolling-modes.png	6.7 KB	10/22/2015	Stanislav Lomany
redraw.png	19.1 KB	10/29/2015	Stanislav Lomany
editing1.png	6.64 KB	12/08/2015	Stanislav Lomany
editing2.png	7.54 KB	12/08/2015	Stanislav Lomany
edit-quirk.png	5.83 KB	12/15/2015	Stanislav Lomany
edit-right-trim.png	1.21 KB	12/18/2015	Stanislav Lomany
deletion.jpg	47.5 KB	01/21/2016	Stanislav Lomany
deletion.p	1.58 KB	01/21/2016	Stanislav Lomany