

User Interface - Feature #2422

add features to BROWSE

10/17/2014 05:25 PM - Greg Shah

<b>Status:</b>	Closed	<b>Start date:</b>	10/17/2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Stanislav Lomany	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	GUI Support for a Complex ADM2 App	<b>version:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			
<b>Related issues:</b>			
Related to Database - Bug #2554: Temp-tables with different labels / column l...		<b>New</b>	<b>04/22/2015</b>
Related to User Interface - Feature #2564: implement GUI BROWSE widget		<b>Closed</b>	
Related to User Interface - Bug #2596: Implement missing browse quirks.		<b>New</b>	<b>06/26/2015</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 #2630: Add ability to save new rows in br...		<b>New</b>	

History

#1 - 10/17/2014 05:36 PM - Greg Shah

Please add (or finish) the following features in BROWSE. At this time, the GUI version of BROWSE is not needed. These features should be fully enabled in ChUI. Any GUI-only features should be implemented to the extent possible. Make notes in this task to document the exact list of any features that are not complete (and cannot be completed without GUI being present).

Options

NO-ASSIGN  
FIT-LAST-COLUMN  
NO-COLUMN-SCROLLING  
ROW-HEIGHT-CHARS  
COLUMN-BGCOLOR  
EXPANDABLE  
OVERLAY  
CENTERED  
SIZE  
SCROLLABLE  
NO-UNDERLINE

Methods

DESELECT-FOCUSED-ROW  
DESELECT-SELECTED-ROW  
DELETE-SELECTED-ROW  
DELETE-SELECTED-ROWS  
DELETE-CURRENT-ROW  
SELECT-ROW  
SELECT-NEXT-ROW  
SELECT-PREV-ROW  
IS-ROW-SELECTED  
INSERT-ROW  
ADD-LIKE-COLUMN  
MOVE-COLUMN  
GET-REPOSITIONED-ROW

Attributes

FIRST-COLUMN  
NEXT-COLUMN  
FOCUSED-ROW  
NUM-LOCKED-COLUMNS  
MIN-HEIGHT-CHARS  
ROW-HEIGHT-CHARS

ALLOW-COLUMN-SEARCHING  
SEPARATORS  
COLUMN-BGCOLOR  
EXPANDABLE  
COLUMN-RESIZABLE  
LABEL-BGCOLOR  
COLUMN-MOVABLE  
FIT-LAST-COLUMN  
SORT  
CURRENT-ROW-MODIFIED  
REFRESHABLE  
NEW-ROW  
MAX-CHARS  
MAX-DATA-GUESS  
CURSOR-OFFSET  
SCROLLBAR-VERTICAL  
CREATE-ON-ADD (undocumented but found in the customer GUI code)  
AUTO-ZAP  
ROW-MARKERS

## #2 - 10/17/2014 05:44 PM - Greg Shah

Please do this work in 2 parts:

1. The conversion and runtime work to get the configuration values set into BrowseConfig or BrowseColumnConfig.
2. The runtime implementation.

The idea is for you to test and check in the changes for part 1 first. That opens up some conversion work for us while you work on part 2.

## #3 - 10/20/2014 01:13 PM - Stanislav Lomany

- Status changed from New to WIP

I'll post here bugs which I'll found along the way.

Bug 1.  
When a value is edited in an editable browse, key presses are not properly propagated. That can be nasty because editor-tab key also doesn't work.

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

```
create tt. tt.f1 = 1. tt.f2 = 11.
create tt. tt.f1 = 2. tt.f2 = 22.
create tt. tt.f1 = 3. tt.f2 = 33.
create tt. tt.f1 = 4. tt.f2 = 44.
create tt. tt.f1 = 5. tt.f2 = 55.
create tt. tt.f1 = 6. tt.f2 = 66.
def buffer xtt for tt.
```

```

def query q for tt.
def browse brws query q
  display tt.f1
  tt.f2
  enable all
    with single 4 down width 70 no-separators
    title "My Browse".

form brws with frame fr.
open query q for each tt.

on tab editor-tab.

enable all with frame fr.

on x,X anywhere do:
  for each xtt:
    message string(xtt.f1) + " " + string(xtt.f2).
  end.
end.

on "1" anywhere do:
  message "1 pressed!".
end.

message "press 123,tab,456,tab,x".

wait-for close of current-window.

```

#### #4 - 10/20/2014 02:26 PM - Greg Shah

OK, good. As you create them, please do check in your testcases to testcases/uast/browse/. This includes bugs like in note 3 above, as well as the standard functional testcases you are writing to explore these features.

#### #5 - 10/30/2014 12:14 PM - Stanislav Lomany

LABEL-BGCOLOR attribute can be applied to a browse or a browse column. Should I create some common interface for BrowseWidget and BrowseColumnWidget so I'll be able to unwrap calls like.

```
some-handle:LABEL-BGCOLOR = 1.
```

to

```
someHandle.unwrapBrowseElement().setLabelBgColor(1);
```

?

**#6 - 10/30/2014 01:12 PM - Greg Shah**

Should I create some common interface for BrowseWidget and BrowseColumnWidget so I'll be able to unwrap calls like.

Yes, this is a good idea.

**#7 - 11/03/2014 03:26 AM - Stanislav Lomany**

- File *svl\_upd20141102a.zip* added

Conversion part for review.

**#8 - 11/03/2014 03:03 PM - Greg Shah**

Code Review 1102a

In general, this is really good.

1. The kw\_dcolor attribute now has a duplicate entry in the methods\_attributes.rules. The hwrap settings are conflicting too.
2. The new attributes and methods should be placed alphabetically in methods\_attributes.rules.
3. I assume you found NO-COLUMN-SCROLLING is a valid attribute through your testcases? Please add KW\_NO\_COLS to the artificial token list in progress.g. I must have neglected that when I added the keyword originally.
4. The \*Element classes are missing the LegacyAttribute annotations for each of the attribute getters/setters.
5. Did you remove expandable from BrowseConfig on purpose?
6. I think that including SORT and MAX-CHARS together in ComboBoxElement will be a problem. SORT can be used on SELECTION-LIST widgets which don't implement ComboBoxElement so the converted code won't unwrap/cast properly. But just adding ComboBoxElement to SelectionListWidget is not right either because MAX-CHARS is not a valid attribute for SELECTION-LIST.

**#9 - 11/03/2014 03:19 PM - Greg Shah**

Code Review 1102a part 2

Looking more carefully, I see that most of the new attributes and methods don't belong to an interface that can be unwrapped. Generally speaking, all methods and attributes need to be placed in an interface and each Java method in that interface that matches a legacy attribute or method should have a LegacyAttribute or LegacyMethod annotation. This interface would then be the unwrapping target instead of BrowseWidget. This approach is needed to allow read-only attribute handling to occur properly at runtime. Look at CommonWidget for an example.

**#10 - 11/05/2014 02:14 PM - Stanislav Lomany**

3. I assume you found NO-COLUMN-SCROLLING is a valid attribute through your testcases?

Yes.

5. Did you remove expandable from BrowseConfig on purpose?

Yes, EXPANDABLE was replaced in Progress by FIT-LAST-COLUMN, so I did the same thing (leaving legacy accessors).

6. I think that including SORT and MAX-CHARS together in ComboBoxElement will be a problem. SORT can be used on SELECTION-LIST widgets which don't implement ComboBoxElement so the converted code won't unwrap/cast properly. But just adding ComboBoxElement to SelectionListWidget is not right either because MAX-CHARS is not a valid attribute for SELECTION-LIST.

I'll create split interfaces SortedElement and MaxCharsElement.

Generally speaking, all methods and attributes need to be placed in an interface and each Java method in that interface that matches a legacy attribute or method should have a LegacyAttribute or LegacyMethod annotation. This interface would then be the unwrapping target instead of BrowseWidget.

I'll create a interface BrowseWidgetInterface, into which I'll put all **new** methods and attributes.

**#11 - 11/05/2014 02:28 PM - Greg Shah**

5. Did you remove expandable from BrowseConfig on purpose?

Yes, EXPANDABLE was replaced in Progress by FIT-LAST-COLUMN, so I did the same thing (leaving legacy accessors).

I wonder if the Progress docs are correct on this point. A question is what Progress does when both expandable and fit-last-column are used on the same browse. And more importantly: does one change when you assign the other?

**#12 - 11/05/2014 02:34 PM - Stanislav Lomany**

A question is what Progress does when both expandable and fit-last-column are used on the same browse.

It is forbidden.

And more importantly: does one change when you assign the other?

Yes.

**#13 - 11/06/2014 08:17 PM - Stanislav Lomany**

- File *svl\_upd20141107a.zip* added

Update for review. Fixed noted flaws and re-merged.

**#14 - 11/07/2014 07:11 AM - Stanislav Lomany**

- File *svl\_upd20141107b.zip* added

Merged with the latest version.

**#15 - 11/07/2014 09:45 AM - Greg Shah**

Code Review svl\_upd20141107b.zip

The changes look good. Please get them regression tested (conversion and runtime).

**#16 - 11/09/2014 11:49 AM - Stanislav Lomany**

Forgot to post. My separation list of attributes that can be implemented for ChUI now or are GUI-only:

NO-ASSIGN - implement  
FIT-LAST-COLUMN - GUI  
NO-COLUMN-SCROLLING - GUI  
ROW-HEIGHT-CHARS - GUI  
COLUMN-BGCOLOR - GUI  
EXPANDABLE - GUI  
OVERLAY - does it have any meaning?  
CENTERED - I found no evidence that it works  
SIZE - already works  
SCROLLABLE - how it works?  
NO-UNDERLINE - implement

DESELECT-FOCUSED-ROW - implement  
DESELECT-SELECTED-ROW - implement  
DELETE-SELECTED-ROW - implement  
DELETE-SELECTED-ROWS - implement  
DELETE-CURRENT-ROW - implement  
SELECT-ROW - implement  
SELECT-NEXT-ROW - implement  
SELECT-PREV-ROW - implement  
IS-ROW-SELECTED - implement  
INSERT-ROW - implement  
ADD-LIKE-COLUMN - Windows (works for ChUI, but browse may have some columns hidden).  
MOVE-COLUMN - GUI

FIRST-COLUMN - implement  
NEXT-COLUMN - implement  
FOCUSED-ROW - implement  
NUM-LOCKED-COLUMNS - implement  
MIN-HEIGHT-CHARS - seems to be a calculated real-only attribute  
ROW-HEIGHT-CHARS - GUI  
ALLOW-COLUMN-SEARCHING - Windows  
SEPARATORS - GUI  
COLUMN-BGCOLOR - looks like implemented  
EXPANDABLE - GUI  
COLUMN-RESIZABLE - GUI  
LABEL-BGCOLOR - GUI  
COLUMN-MOVABLE - GUI  
FIT-LAST-COLUMN - GUI  
SORT - implement  
CURRENT-ROW-MODIFIED - implement  
REFRESHABLE - implement  
NEW-ROW - implement  
MAX-CHARS - GUI  
MAX-DATA-GUESS - implement  
CURSOR-OFFSET - implement  
SCROLLBAR-VERTICAL - GUI  
CREATE-ON-ADD - implement  
AUTO-ZAP - implement  
ROW-MARKERS - implement

**#17 - 11/10/2014 09:48 AM - Greg Shah**

Good summary.

How is regression testing going? Eugenie and I both have updates dependent upon yours, so it will be helpful to get yours checked in soon.

**#18 - 11/10/2014 10:24 AM - Stanislav Lomany**

Conversion part is OK, runtime is still not complete, it hung last time.

**#19 - 11/11/2014 05:03 AM - Stanislav Lomany**

svl\_upd20141107b.zip was committed to bzt rev 10651.

**#20 - 11/11/2014 09:38 AM - Greg Shah**

I realized that I missed something in the code review. All the new attribute/method interfaces needed to be added to the special HandleCommon aggregator interface. I have a large number of changes which I just merged up with your bzt rev 10651. I have changes in HandleCommon as well, so I am putting all your new interfaces into that file as well. This is just FYI so that in the future you'll know to make that change too.

How is the runtime work going?

**#21 - 11/11/2014 09:44 AM - Stanislav Lomany**

I decided to finish with the MAJIC task first.

**#22 - 11/14/2014 08:05 AM - Stanislav Lomany**

Can I have a look at the code which uses NO-ASSIGN option? This option is intended for use with user-defined triggers on the ROW-LEAVE event, so I want to ensure that I will not affect functionality that saves a record from the trigger.

**#23 - 11/14/2014 08:50 AM - Greg Shah**

Please see ~/secure/clients/<customer\_name>/reports/<app\_name>\_cut\_down\_gui\_reports\_2010A08\_06\_20141015.zip. Inside, look for the "Browse Options" report. NO-ASSIGN is used 521 times. Click through to see the source code (.cache files) that use the option.

LE: GES removed the customer's name and the application name from this entry.

**#24 - 11/17/2014 02:27 PM - Stanislav Lomany**

Bug 2.

If an editable browse has another browse as a sibling, it is practically unusable.

See uast/browse/brws-no-underline.p

**#25 - 11/17/2014 06:00 PM - Greg Shah**

Stanislav Lomany wrote:

Bug 2.

If an editable browse has another browse as a sibling, it is practically unusable.

See uast/browse/brws-no-underline.p

Is this a P2J bug or a strange behavior of the 4GL?



## #26 - 11/18/2014 05:45 AM - Stanislav Lomany

It is P2J bug(s).

## #27 - 11/20/2014 11:34 AM - Stanislav Lomany

(Greg) Here is a list of those features needed:

methods DESELECT-FOCUSED-ROW, DESELECT-SELECTED-ROW, DELETE-CURRENT-ROW, SELECT-ROW, IS-ROW-SELECTED;  
attributes FIRST-COLUMN, NEXT-COLUMN, FOCUSED-ROW, NUM-LOCKED-COLUMNS  
browse options OVERLAY, CENTERED, SCROLLABLE, NO-UNDERLINE

To the degree that you have found no behavior for a feature (e.g. CENTERED), make sure you have tried enough variations to be reasonably confident that we don't need it. It is possible that this is just a quirk of the 4GL parsing since much of the same syntax is allowed for both frame options (the frame phase) and browse options. Both phrases are implemented using the WITH keyword. So it is possible that the syntax is allowed but silently ignored. On the other hand, I don't want to miss something important and then have delays while we figure things out.

I can also provide examples of how these are used. For CENTERED and OVERLAY:

```
DEFINE TEMP-TABLE ttSelection
    FIELD iElementId AS INTEGER
    FIELD cMoniker AS CHAR
    FIELD cAddr AS CHAR
    FIELD lFullAddr AS LOGICAL.

DEFINE QUERY qSel          FOR ttSelection SCROLLING .

DEFINE BROWSE brSel QUERY qSel
    DISPLAY
        ttSelection.cAddr FORMAT "x(74)"
        WITH 10 DOWN CENTERED ROW 3 NO-LABELS OVERLAY NO-BOX.

DEFINE FRAME frSel
    brSel
        HELP "Arrow To Selection And Press F1 To Select Suggested Address"
        WITH CENTERED OVERLAY ROW 3
        TITLE " Suggested Addresses ".
```

## #28 - 11/20/2014 11:53 AM - Stanislav Lomany

I think that OVERLAY, CENTERED and SCROLLABLE options are silently ignored.

OVERLAY - I thought that it might be used to work around some 4GL behavior for overlay frames, but everything works as expected.  
CENTERED - just doesn't make browse centered.  
SCROLLABLE - I haven't seen an example, but I haven't noticed any differences in scrolling behavior. And I have no idea how it might work.

#### #29 - 11/25/2014 04:15 PM - Stanislav Lomany

There is 4GL quirk when SELECT-ROW or some functions are used and the target browse doesn't have the focus, after there functions have been applied, the current browse row is highlighted as if the browse is focused (while it's not). That is ChUI-only quirk. Should I emulate it?

#### #30 - 11/25/2014 04:24 PM - Greg Shah

That is ChUI-only quirk. Should I emulate it?

If it will take days to do, then defer it (but create a new task and make it related to this one so we don't forget).

If you can do it in hours, then go ahead.

#### #31 - 11/26/2014 08:30 AM - Stanislav Lomany

Bug 3. About editable browse. Testcase browse/brws-select3-edit.p.

Reproduction: press F4.

```
java.lang.NullPointerException
    at com.goldencode.p2j.ui.client.Browse.dcGetRowIndex(Browse.java:3078)
    at com.goldencode.p2j.ui.client.Browse.dcGetRow(Browse.java:2033)
    at com.goldencode.p2j.ui.client.Browse.updateColumn(Browse.java:1819)
    at com.goldencode.p2j.ui.client.Browse.stopEdit(Browse.java:1795)
    at com.goldencode.p2j.ui.client.Browse.refresh(Browse.java:2845)
    at com.goldencode.p2j.ui.client.Browse.queryClosed(Browse.java:2931)
    at com.goldencode.p2j.ui.chui.ThinClient$22.run(ThinClient.java:7180)
    at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:11781)
    at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:11730)
    at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:11670)
    at com.goldencode.p2j.ui.chui.ThinClient.queryClosed(ThinClient.java:7174)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:257)
    at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1128)
    at com.goldencode.p2j.net.Queue.transact(Queue.java:585)
    at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:223)
    at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:163)
    at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1423)
    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:285)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:94)
    at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:227)
    at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:423)
    at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:112)
    at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:290)
```

Then press F4 one more time and the following stacktrace will repeatedly appear until client got killed.

```
java.lang.reflect.InvocationTargetException
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.Utills.invoke(Utills.java:1244)
    at com.goldencode.p2j.main.StandardServer$MainInvoker.execute(StandardServer.java:1701)
    at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1201)
    at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:364)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
    at java.lang.Thread.run(Thread.java:744)
Caused by: java.lang.RuntimeException: A configuration is already associated for 4: fl 4 com.goldencode.p2j.ui
.client.Dimension@bca2b87 com.goldencode.p2j.ui.client.Point@5efb21d visible false can false;
    at com.goldencode.p2j.ui.ConfigManager.addWidgetConfig(ConfigManager.java:381)
    at com.goldencode.p2j.ui.client.WidgetRegistry.reconstructWidget(WidgetRegistry.java:110)
    at com.goldencode.p2j.ui.client.WidgetRegistry.pushDefinition(WidgetRegistry.java:325)
    at com.goldencode.p2j.ui.chui.ThinClient$17.run(ThinClient.java:6664)
    at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:11781)
    at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:11730)
    at com.goldencode.p2j.ui.chui.ThinClient.pushOneDef(ThinClient.java:6652)
    at com.goldencode.p2j.ui.chui.ThinClient.pushScreenDefinition(ThinClient.java:6627)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:257)
    at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1128)
    at com.goldencode.p2j.net.Queue.transact(Queue.java:585)
    at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:223)
    at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:163)
    at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1423)
    at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:97)
    at com.sun.proxy.$Proxy4.standardEntry(Unknown Source)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:285)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:94)
    at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:227)
    at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:423)
    at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:112)
    at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:290)
[11/26/2014 16:17:52 FET] (StandardServer.invoke:SEVERE) {00000004:00000013:bogus} Abnormal end!
java.lang.reflect.InvocationTargetException
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.Utills.invoke(Utills.java:1244)
    at com.goldencode.p2j.main.StandardServer$MainInvoker.execute(StandardServer.java:1701)
    at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1201)
    at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:364)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
    at java.lang.Thread.run(Thread.java:744)
Caused by: java.lang.IllegalArgumentException: Invalid cache read request: 0. Cache contains only [0:0)
    at com.goldencode.p2j.ui.client.Browse.dcGetRowIndex(Browse.java:3080)
    at com.goldencode.p2j.ui.client.Browse.dcGetRow(Browse.java:2033)
```

```
at com.goldencode.p2j.ui.client.Browse.getValue(Browse.java:2724)
at com.goldencode.p2j.ui.client.Browse.startEdit(Browse.java:1752)
at com.goldencode.p2j.ui.client.Browse.setReadOnly(Browse.java:861)
at com.goldencode.p2j.ui.BrowseConfig.syncWithWidget(BrowseConfig.java:253)
at com.goldencode.p2j.ui.ConfigManager.replaceConfig(ConfigManager.java:333)
at com.goldencode.p2j.ui.client.Browse.initialize(Browse.java:431)
at com.goldencode.p2j.ui.client.WidgetRegistry.reconstructWidget(WidgetRegistry.java:106)
at com.goldencode.p2j.ui.client.WidgetRegistry.pushDefinition(WidgetRegistry.java:325)
at com.goldencode.p2j.ui.chui.ThinClient$17.run(ThinClient.java:6664)
at com.goldencode.p2j.ui.chui.ThinClient.eventBracket(ThinClient.java:11781)
at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket(ThinClient.java:11730)
at com.goldencode.p2j.ui.chui.ThinClient.pushOneDef(ThinClient.java:6652)
at com.goldencode.p2j.ui.chui.ThinClient.pushScreenDefinition(ThinClient.java:6627)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
at com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:257)
at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1128)
at com.goldencode.p2j.net.Queue.transact(Queue.java:585)
at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:223)
at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:163)
at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1423)
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:285)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:94)
at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:227)
at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:423)
at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:112)
at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:290)
```

### #32 - 11/27/2014 03:55 AM - Stanislav Lomany

The bug in note 3 was bringing some difficulties to my testcases, so I made a fix / workaround:

Was (EventManager):

```
private TriggerResult invokeTriggers(Widget widget,
                                     int      evtCode,
                                     boolean  isKey)
{
    TriggerResult tr = new TriggerResult();

    if (widget != null && widget.config() != null && WidgetId.virtualWidget(widget.config().id))
    {
        // protection against virtual sources
        return tr;
    }
}
```

Become:

```
private TriggerResult invokeTriggers(Widget widget,
                                     int      evtCode,
                                     boolean  isKey)
{
    TriggerResult tr = new TriggerResult();

    if (widget != null && widget.config() != null && WidgetId.virtualWidget(widget.config().id))
    {
        if (widget instanceof FillIn)
            widget = ((FillIn) widget).getBrowse();
        else
            return tr;
    }
}
```

Also add this method to FillIn:

```
public Browse<O> getBrowse()
{
    return browse;
}
```

Since I'm not very familiar with key processing, it would be nice if some could evaluate how close it is to the true solution. On my side I haven't found any glitches during its exploitation.

### #33 - 11/27/2014 05:05 AM - Stanislav Lomany

Bug 4. Missing selected row pointer in single-selection editable browse. Testcase uast/browse/brws-select1-edit.p. Reproduction: focus on the right browse, D, TAB. The selected row pointer ">" is missing in the left browse while the row is actually selected. You can check its state by pressing S. The issue is related to some deep ChUI processing.

### #34 - 11/27/2014 12:17 PM - Greg Shah

```
if (widget instanceof FillIn)
    widget = ((FillIn) widget).getBrowse();
else
    return tr;
```

I don't think this will even compile since it doesn't return anything in the FillIn path. Your version must be different.

Also, this doesn't seem right since it would seemingly disable trigger processing for all fill-ins that are not in browses.

### #35 - 11/27/2014 04:21 PM - Stanislav Lomany

This bug (from note 3) was fixed in the latest release. Great!

### #36 - 11/30/2014 03:30 PM - Stanislav Lomany

Bug 5. Mismatched highlighted and actual current row for editable browse. Testcase: uast/browse/brws-select1-edit-single2.p  
Reproduction: make the button focused, v, 5, enter. First, you can see that the value of the select row disappears, which is a bug. Then press tab to return to the browse. Row 1 is enabled for editing, while the "current row" is the row number 5. You can check that by pressing "up" - row number 4 is selected.

### #37 - 12/02/2014 07:35 AM - Stanislav Lomany

Bug 6. Testcase: uast/browse/brws-refresh.p. Reproduction: press "r" six times. On each press current row goes down, which is incorrect. After the last press we get this:

```
Caused by: java.lang.IllegalArgumentException: Invalid cache read request: 9. Cache contains only [0:9)
    at com.goldencode.p2j.ui.client.Browse.dcGetRowIndex(Browse.java:3080)
    at com.goldencode.p2j.ui.client.Browse.dcGetRow(Browse.java:2033)
    at com.goldencode.p2j.ui.chui.BrowseImpl.access$1400(BrowseImpl.java:39)
    at com.goldencode.p2j.ui.chui.BrowseImpl$DataColumnRenderer.newRow(BrowseImpl.java:707)
    at com.goldencode.p2j.ui.chui.BrowseImpl.draw(BrowseImpl.java:218)
    at com.goldencode.p2j.ui.client.Browse.refresh(Browse.java:2894)
    at com.goldencode.p2j.ui.client.Browse.refresh(Browse.java:2803)
    at com.goldencode.p2j.ui.chui.ThinClient.refresh(ThinClient.java:7053)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
    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$RemoteAccess.invokeCore (RemoteObject.java:1423)
at com.goldencode.p2j.net.InvocationStub.invoke (InvocationStub.java:97)
at com.sun.proxy.$Proxy2.trigger (Unknown Source)
at com.goldencode.p2j.ui.chui.ThinClient.trigger (ThinClient.java:10173)
at com.goldencode.p2j.ui.chui.ThinClient.invokeTriggers (ThinClient.java:13940)
at com.goldencode.p2j.ui.chui.ThinClient.processProgressEvent (ThinClient.java:13335)
at com.goldencode.p2j.ui.chui.ThinClient.processEventsWorker (ThinClient.java:12880)
at com.goldencode.p2j.ui.chui.ThinClient.pop (ThinClient.java:11961)
at com.goldencode.p2j.ui.chui.ThinClient.eventBracket (ThinClient.java:11944)
at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket (ThinClient.java:11870)
at com.goldencode.p2j.ui.chui.ThinClient.applyWorker (ThinClient.java:11664)
at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker (ThinClient.java:9618)
at com.goldencode.p2j.ui.chui.ThinClient.waitFor (ThinClient.java:9161)
at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke (Method.java:606)
at com.goldencode.p2j.util.MethodInvoker.invoke (MethodInvoker.java:76)
at com.goldencode.p2j.net.Dispatcher.processInbound (Dispatcher.java:693)
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$RemoteAccess.invokeCore (RemoteObject.java:1423)
at com.goldencode.p2j.net.InvocationStub.invoke (InvocationStub.java:97)
at com.sun.proxy.$Proxy4.standardEntry (Unknown Source)
at com.goldencode.p2j.main.ClientCore.start (ClientCore.java:286)
at com.goldencode.p2j.main.ClientCore.start (ClientCore.java:95)
at com.goldencode.p2j.main.ClientDriver.start (ClientDriver.java:227)
at com.goldencode.p2j.main.CommonDriver.process (CommonDriver.java:423)
at com.goldencode.p2j.main.ClientDriver.process (ClientDriver.java:112)
at com.goldencode.p2j.main.ClientDriver.main (ClientDriver.java:290)

```

After that this stacktrace pops up continuously until the client is killed:

```

[12/02/2014 15:25:54 FET] (StandardServer.invoke:SEVERE) {00000002:0000000D:bogus} Abnormal end!
java.lang.reflect.InvocationTargetException

```

```

at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke (Method.java:606)
at com.goldencode.p2j.util.Utils.invoke (Utils.java:1244)
at com.goldencode.p2j.main.StandardServer$MainInvoker.execute (StandardServer.java:1701)
at com.goldencode.p2j.main.StandardServer.invoke (StandardServer.java:1201)
at com.goldencode.p2j.main.StandardServer.standardEntry (StandardServer.java:364)
at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke (Method.java:606)
at com.goldencode.p2j.util.MethodInvoker.invoke (MethodInvoker.java:76)
at com.goldencode.p2j.net.Dispatcher.processInbound (Dispatcher.java:693)
at com.goldencode.p2j.net.Conversation.block (Conversation.java:319)
at com.goldencode.p2j.net.Conversation.run (Conversation.java:163)
at java.lang.Thread.run (Thread.java:744)

```

Caused by: java.lang.NullPointerException

```

at com.goldencode.p2j.ui.client.chui.driver.ChuiOutputManager.activateWindow (ChuiOutputManager.java:11

```

3)

```

at com.goldencode.p2j.ui.client.OutputManager.setInvalidate (OutputManager.java:1184)
at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket (ThinClient.java:11875)
at com.goldencode.p2j.ui.chui.ThinClient.pushOneDef (ThinClient.java:6693)
at com.goldencode.p2j.ui.chui.ThinClient.pushScreenDefinition (ThinClient.java:6668)
at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke (Method.java:606)
at com.goldencode.p2j.util.MethodInvoker.invoke (MethodInvoker.java:76)
at com.goldencode.p2j.net.Dispatcher.processInbound (Dispatcher.java:693)
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$RemoteAccess.invokeCore(RemoteObject.java:1423)
at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:97)
at com.sun.proxy.$Proxy4.standardEntry(Unknown Source)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:286)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:95)
at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:227)
at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:423)
at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:112)
at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:290)
```

I have to fix this bug, otherwise DELETE-CURRENT-ROW will not work properly.

**#38 - 12/02/2014 09:06 AM - Greg Shah**

I have to fix this bug, otherwise DELETE-CURRENT-ROW will not work properly.

Understood.

When do you think you will have this next set of functionality ready for review?

**#39 - 12/02/2014 09:38 AM - Stanislav Lomany**

It might be that this bug is the last thing that holds me, so my estimate is from 2 days to until the end of the week in the worst case.

**#40 - 12/10/2014 09:45 AM - Stanislav Lomany**

- File `svl_upd20141210a.zip` added

Update for review. Contains functions listed in note 27 except NUM-LOCKED-COLUMNS.

**#41 - 12/12/2014 10:13 AM - Stanislav Lomany**



About NUM-LOCKED-COLUMNS. It may be the case for an editing browse that available (non-locked) width is less than the width of a scrollable column (or you can make simple browse narrow enough so a column cannot entirely fit into it). In this case editing is buggy. In 4GL it is buggy and practically unusable too.

**#42 - 12/15/2014 01:56 PM - Greg Shah**

Code Review svl\_upd20141210a.zip

The changes look very good.

When can I expect the final update? I'd like to get it tested and checked in ASAP.

**#43 - 12/15/2014 05:36 PM - Stanislav Lomany**

There is a minor issue for editable browse remains, I expect to fix it and send the update tomorrow.

**#44 - 12/16/2014 11:09 AM - Stanislav Lomany**

- File svl\_upd20141216a.zip added

Update for review. Implements NUM-LOCKED-COLUMNS. Based on svl\_upd20141210a.zip.

**#45 - 12/16/2014 11:42 AM - Greg Shah**

Code Review svl\_upd20141216a.zip

1. Browse, BrowseImpl and BrowseWidgetb should only have a single history entry for all of your changes.
2. There are quite a few missing files that had been included with svl\_upd20141210a.zip.

**#46 - 12/16/2014 11:45 AM - Greg Shah**

After you have this update finalized and into testing, please work on the rest of the non-GUI features in the original list.

**#47 - 12/16/2014 04:00 PM - Stanislav Lomany**

I'll just put here the list of remaining features to implement:

NO-ASSIGN - implement

DELETE-SELECTED-ROW - implement

DELETE-SELECTED-ROWS - implement

SELECT-NEXT-ROW - implement

SELECT-PREV-ROW - implement

INSERT-ROW - implement

ADD-LIKE-COLUMN - works for ChUI to a certain extent

MIN-HEIGHT-CHARS - seems to be a calculated real-only attribute

COLUMN-BGCOLOR - looks like implemented

SORT - implement

CURRENT-ROW-MODIFIED - implement

REFRESHABLE - implement

NEW-ROW - implement

MAX-DATA-GUESS - implement

CURSOR-OFFSET - implement

CREATE-ON-ADD - implement

AUTO-ZAP - implement

ROW-MARKERS - implement

**#48 - 12/19/2014 08:29 AM - Stanislav Lomany**

About NO-ASSIGN option - in 4GL when a value wasn't assigned, the cell screen value is reverted to its database value when the row goes out of the view box. In P2J value is reverted as soon as we perform vertical scrolling. That means 4GL caches on-screen records. Should I reproduce this behavior?

**#49 - 12/19/2014 08:42 AM - Greg Shah**

Stanislav Lomany wrote:

About NO-ASSIGN option - in 4GL when a value wasn't assigned, the cell screen value is reverted to its database value when the row goes out of the view box. In P2J value is reverted as soon as we perform vertical scrolling. That means 4GL caches on-screen records. Should I reproduce this behavior?

Yes, please do implement this. I thought we already did cache the on screen records, but I guess we don't do it quite right.

This only occurs when NO-ASSIGN is specified?

**#50 - 12/19/2014 10:49 AM - Stanislav Lomany**

No, the same effect may be achieved by updating browse records in background.

**#51 - 12/19/2014 11:25 AM - Greg Shah**

Stanislav Lomany wrote:

No, the same effect may be achieved by updating browse records in background.

OK. Please implement the full support for this behavior.

**#52 - 12/19/2014 02:29 PM - Stanislav Lomany**

- File *svl\_upd20141218a.zip* added

Just for the case, please review the update which passed regression testing so I can check it in.

**#53 - 12/19/2014 03:17 PM - Greg Shah**

Code Review *svl\_upd20141218a.zip*

It looks good. Please do check it in.

**#54 - 12/19/2014 04:02 PM - Stanislav Lomany**

svl\_upd20141218a.zip was committed into bzt rev 10684.

**#55 - 12/22/2014 04:25 PM - Stanislav Lomany**

For a note: there is a number of issues related to how records are loaded into browse, for example:

1. P2J caches more records than it should.
  - 1.1. For adaptive queries getting query size forces rows to be cached by query cursor earlier than it should.
  - 1.2. P2J gets rows for browse in 3-page chunks (why?).
2. When a record is deleted in background is it not properly handled by browse/query and browse navigation may stuck on that record.

I'm working on retaining currently viewed records in the browse cache. And in paged caching approach (P2J) it is implemented differently than in incremental (4GL, get only records you need). Obviously complete analysis and modification of the core of browse caching mechanism is an effort-consuming task. What do you think?

BTW, PageDown stops working after several pages. I'll fix it.

**#56 - 12/22/2014 07:56 PM - Greg Shah**

Interesting findings.

It seems to me that some basic differences in the core caching mechanism could lead to many hard to find and subtle problems.

Now is the time to fix these problems. We have important projects that are going to push the boundaries of our browse support. I want to proactively cleanup and extend our browse support now so that we don't have to spend many months debugging later, especially since "later" may be at a time when this code will be in production. Fixing it now should be less effort and less stress in the long run.

To the degree that the changes involve the query infrastructure or the persistence layer, Eric will have to review and approve those changes. I'll be involved for the UI side.

**#57 - 12/24/2014 08:28 AM - Stanislav Lomany**

I was intended to check if BROWSE browsing is based on row numbers as it is implemented in P2J (not, say, on record ids). The plan was:

1. open a transaction;
2. delete a record;
3. load a view which contains adjacent records of the deleted record;
4. undo the transaction.

For standalone queries performing similar trick doesn't affect row numbering (deleted rows occupy row numbers). But for browse it turns out that deleted (even if deletion is not committed) records are immediately deleted from the result list if they failed to load into a view and row numbers are shifted.

**#58 - 12/27/2014 05:38 PM - Stanislav Lomany**

Some thoughts:

- Uncommitted deletions are deleted from the result list if records failed to load into a view. Row numbers are updated immediately.
- For committed deletions row numbers are not updated unless you explicitly REPOSITION bound query to the deleted row. After that row numbers are updated.
- DELETE/CREATE-RESULT-LIST-ENTRY and DELETE-SELECTED-ROWS update row numbers immediately.
- Windows version of Progress has bugs - in case of record deletion cursor may jump, some extra records may be missing, selection may be invalid. ChUI version is more predictable, but it has some bugs too. So I think that our mission is to create a predictable browse which mimics Progress as much as it makes sense.
- It terms of result list modification we have: 1. Row deletions/insertions with immediate row numbers update, most of which are initiated by browse. 2. Row deletions without updating row numbers if record cannot be loaded (but eventually these gaps of unused row numbers can be closed).  
I think that we can live with existing P2J model which assumes: 1. Row numbers are updated immediately after DELETE-RESULT-LIST-ENTRY/DELETE-SELECTED-ROWS. 2. Row numbers are not updated at all if a record cannot be loaded.
- Note that for multiple selection case we have to be careful and make sure that we point to the same rows even if the result list has changed.

Considering all of the above I suggest to cache rows on the client side together with their row numbers. Normally row numbers are continuous, but potentially there may be gaps. In order to scroll to specific direction I suggest to 1. Reposition query to the top/bottom row in the view (we got row numbers from the server) 2. Get as many records as it is required to build the next view using the cached records and the records retrieved using next/previous operations (note the current P2J implementation uses only "next" operation).

What do you think?

**#59 - 12/29/2014 08:14 AM - Greg Shah**

Windows version of Progress has bugs - in case of record deletion cursor may jump, some extra records may be missing, selection may be invalid. ChUI version is more predictable, but it has some bugs too. So I think that our mission is to create a predictable browse which mimics Progress as much as it makes sense.

For NOW, we can try this approach. BUT, in the past we have often found application code that relies upon these bugs which then turns the bugs into features which we do need to support exactly.

We must get confirmation about the bugs and be prepared to replicate that behavior. Please document the exact behavior in each found bug which you propose to leave un-replicated. Highlight the difference in behavior from that which you plan to leave in P2J. We will then present that list to the customer for review. If they confirm that the bugs are things that they don't care about, then we will go with that approach. If they have concerns then we will implement those features that they believe are important.

Considering all of the above I suggest to cache rows on the client side together with their row numbers. Normally row numbers are continuous, but potentially there may be gaps. In order to scroll to specific direction I suggest to 1. Reposition query to the top/bottom row in the view (we got row numbers from the server) 2. Get as many records as it is required to build the next view using the cached records and the records retrieved using next/previous operations (note the current P2J implementation uses only "next" operation).  
What do you think?

This sounds reasonable to me, but I think Eric really needs to review the idea as he will understand the implications better.

Eric?

**#60 - 12/29/2014 01:15 PM - Eric Faulhaber**

Greg Shah wrote:

Considering all of the above I suggest to cache rows on the client side together with their row numbers. Normally row numbers are continuous, but potentially there may be gaps. In order to scroll to specific direction I suggest to 1. Reposition query to the top/bottom row in the view (we got row numbers from the server) 2. Get as many records as it is required to build the next view using the cached records and the records retrieved using next/previous operations (note the current P2J implementation uses only "next" operation). What do you think?

This sounds reasonable to me, but I think Eric really needs to review the idea as he will understand the implications better.

Eric?

I'm generally OK with the approach, but I'm concerned about memory use. Do cached rows remain cached for the life of the browse, or is the idea to expire any rows from the cache eventually? Could a user browsing a huge query get us into trouble by scrolling down for a very long time, causing the cache to grow to an unreasonable size? Any idea how the 4GL deals with this?

**#61 - 12/29/2014 01:55 PM - Stanislav Lomany**

Eric, I'm not going to cache all rows, I'm only going to additionally cache row numbers and number of rows + row numbers is limited to the view size.

**#62 - 12/29/2014 04:47 PM - Eric Faulhaber**

OK, sounds good to me.

**#63 - 01/12/2015 12:29 PM - Stanislav Lomany**

Here is the list of 4GL bugs which I'm not going to reproduce. It's obviously not full and may expand it as I'll find new ones. All reproductions are for uast/browse/brws-no-assign3.p

1. Invalid current row when passing by a record deleted outside the view. (UI only)

```
Down to record 120
X, 50, Enter, 60, Enter (delete records 50 and 60)
Up 6 times - the current row is 90, and the scroll row 40, while the current row should also be 40.
```

2. Invalid selection after a record was deleted outside the view. (UI only)

```
Select records 10-80
Down to record 120
X, 40, Enter, 60, Enter (delete records 40, 50 and 60)
Scroll up to check the selection - it is records 10, 70, 80, 100
```

### 3. Invalid selection after reposition. (UI / ChUI)

```
Select records 10-80
Down to record 120
D, 60, Enter (delete record 60)
Q, 4, Enter (reposition to row 4)
Scroll to check the selection - it is records 10, 20, 30, 40, 50, 70, 130 (ChUI) or 120 (UI)
```

### 4. Navigation is stopped after leading records has been deleted. (UI only)

```
X, 10, Enter, 20, Enter (delete records 10 and 20)
Down to record 60
Up to 30
Up once more - current row highlighting has disappeared, you cannot scroll up/down anymore. But you can select
a row using mouse to continue using browse normally.
```

### #64 - 01/20/2015 05:59 PM - Stanislav Lomany

Some differences in handling of Page Up/Down in ChUI and UI versions.

1. About Page Down key (in ChUI it's not actually Page Down, but some weird key combination):  
In ChUI it shows the next page, current row position in the viewport remains the same. The last page can be incomplete.  
In UI it increments the current row with page size - 1 and places the new current row in the bottom of the view. The last page cannot be incomplete.  
Page Up works in the similar way.
2. In order to get the previous page, ChUI version decrements the scroll row by the page size. I.e. if some rows on the previous page are deleted you will see remaining row on that page appended with the rows from the current page. If the previous page is fully deleted, you have to press Page Up twice in order to move to the page before the deleted one.  
ChUI version is too buggy in this case, but it look like it tries to get enough rows to perform the full scroll. I.e. if some rows on the previous page are deleted, it will take rows from the following previous pages.  
When it comes to getting the next page, both versions work similar trying to completely fill the browse with available following rows.

For now I'll stick to ChUI version.

#### #65 - 01/21/2015 08:14 AM - Greg Shah

You'll be working on GUI soon, so do make sure that your approach can be easily adapted for those cases.

#### #66 - 01/21/2015 08:57 AM - Stanislav Lomany

About this particular feature:

If the previous page is fully deleted, you have to press Page Up twice in order to move to the page before the deleted one.

This requires implementation of the mechanism that closes gaps of unused row numbers upon visiting these rows. Which IMO brings unnecessary complications, so I'm not going to implement this feature.

#### #67 - 02/02/2015 03:55 PM - Stanislav Lomany

Addition to note 63. Another 4GL bug which I'm not going to reproduce. Applies to ChUI and GUI. If the number of rows is less or equals the page size then browse is not scrolled upon query reposition ("mode 1"), otherwise current row becomes the scroll (top) row ("mode 2"). however if there were more records than the page size, then some records were deleted and the number of records equals page size that brings the browse to inconsistent state.

Reproduction (brws-no-assign3.p):

```
x, 50, enter, 1000, enter
q, 2, enter
```

Browse is scrolled to row 20 and row 30 is selected.

Another reproduction

```
x, 50, enter, 1000, enter
q, 4, enter
down arrow
```

Error in umBrowseGetNextRow (456) is displayed, you cannot scroll browse.

I'm going to implement the following logic:

1. Fetch one extra row on the first fetch to determine if the number of rows exceeds the page size (4GL does this).
2. 4GL handles records deletions by closing gaps of deleted records. So, if some records were deleted and the number of records become less than the page size then, depending on how we use browse, it may eventually switch from "mode 2" to "mode 1". I think we can improve browse behavior in this respect by switching to "mode 1" if the record number 1 is in the view and the number of rows is less than page size.

#68 - 02/09/2015 11:07 AM - Stanislav Lomany

It turns out that for ChUI behavior of Page Down differs for preselect and adaptive queries. For example the following sequence down, down, i, x, 70, enter, 1000, enter, i leads to brws-no-assign3.p:

```
x 60
. .
. .
. .
```

brws-no-assign3a.p:

```
50
60
70
x 80
```

I decided to implement only adaptive way of behavior for now because:

- 1. it is more common in Majic;
- 2. it uses more natural logic;
- 3. it is closer to GUI behavior;
- 4. I do not yet completely understand how preselect case works, it's not that simple, and IMO trying to completely reproduce it is a time waste.

#69 - 02/09/2015 11:22 AM - Stanislav Lomany

Addition to note 63. Another 4GL bug which I'm not going to reproduce. Applies to GUI only. Reproduction: down to 80, up to 60, x, 70, enter, 1000, enter, Page Down \* 3  
Output:

```
50
60
50
x 60
```



**#70 - 02/16/2015 03:32 PM - Stanislav Lomany**

FYI, Majic alters browse column "colors" in two places. E.g.:

```
assign w-mor.priority:dcolor in browse mor-brws = 4.
```

while BrowseColumnWidget.setDColor(NumberType) doesn't actually set a color because ColorSpec.convert() always returns default color (at least for me).

But I can handle browsing part because BrowseColumnWidget.setDColor(Color) does the job.

**#71 - 02/16/2015 10:31 PM - Greg Shah**

while BrowseColumnWidget.setDColor(NumberType) doesn't actually set a color because ColorSpec.convert() always returns default color (at least for me).

Unless we configure a set of Linux terminal color mappings, the number 4 would always return the default.

But I can handle browsing part because BrowseColumnWidget.setDColor(Color) does the job.

I don't understand what you mean here.

What has changed that you are trying to resolve?

**#72 - 02/17/2015 02:06 AM - Stanislav Lomany**

But I can handle browsing part because BrowseColumnWidget.setDColor(Color) does the job.

I don't understand what you mean here.

I found a way to work with colors without setting terminal color mappings.

What has changed that you are trying to resolve?

- 1. Coloring wasn't properly working.
- 2. Since color data is a part of caching structures and caching mechanism has changed, I have to modify it anyway.

#73 - 02/20/2015 11:25 AM - Stanislav Lomany

Technically that's not a browse problem, but because of record retrieval logic browse uses, an editable browse easily exposes this bug. Assigning a value to a field of the index which drives an adaptive query for a browse discards not yet loaded records into browse which are now positioned before the updated record.

Testcase:

```
def temp-table tt field f1 as integer
    index idx f1.

def var i as integer.
repeat i = 1 to 10:
    create tt. tt.f1 = i.
end.

def query q for tt scrolling.
open query q for each tt.

form tt.f1 with frame f1 15 down.

repeat:
    get next q.
    if not avail(tt) then leave.

    if tt.f1 = 5 then do transaction:
        tt.f1 = 9999.
        display tt.f1 with frame f1.
        down with frame f1.

        /* That refreshes the record and cleans buffer's snapshot */
        reposition q to row 5.
    end.
    else do:
        display tt.f1 with frame f1.
        down with frame f1.
    end.

end.
```

4GL:

f1
1
2
3
4
9,999
9,999
6
7
8
9
10
9,999

--

P2J:

f1
1
2
3
4
9,999
9,999

I'm not sure what can be done with the logic of the adaptive query which emulates 4GL index walking. Any ideas?

**#74 - 02/25/2015 05:53 AM - Stanislav Lomany**

- File svl\_upd20150225a.zip added

Major browse update for review. Changed caching logic in order to match 4GL.

**#75 - 02/26/2015 09:38 AM - Greg Shah**

Code Review svl\_upd20150225a.zip

I'm fine with the changes. I looked at everything pretty carefully, except Browse.java which has so many changes that I could not read every line. I skimmed the changes in that file and did not see anything obviously wrong.

Eric: please review the persistence changes.

If Eric accepts the changes and they pass regression testing, you can check them in.

**#76 - 02/26/2015 12:09 PM - Eric Faulhaber**

The persistence changes in 0225a look OK to me.

**#77 - 02/26/2015 12:15 PM - Stanislav Lomany**

After my recent minor changes RFQ tests started to fail. I'm looking into that.

**#78 - 02/26/2015 12:25 PM - Eric Faulhaber**

Stanislav Lomany wrote:

After my recent minor changes RFQ tests started to fail. I'm looking into that.

I would suspect the QueryWrapper change. This is what I was seeing when I tested this change on its own, but I thought this was passing when you tested it with your other changes.

**#79 - 03/02/2015 03:47 PM - Stanislav Lomany**

- File svl\_upd20150302a.zip added

Update for review with fixes for RFQ tests. Regression testing is running.

**#80 - 03/02/2015 05:05 PM - Stanislav Lomany**

Passed regression testing, so I can check it in just after review.

**#81 - 03/02/2015 05:11 PM - Greg Shah**

Code Review svl\_upd20150302a.zip

I'm OK with the changes.

Eric: please review the AdaptiveQuery change.

**#82 - 03/02/2015 09:29 PM - Eric Faulhaber**

Greg Shah wrote:

Code Review svl\_upd20150302a.zip

I'm OK with the changes.

Eric: please review the AdaptiveQuery change.

Stas, you removed this code from AdaptiveQuery:

```
// Emulate a quirk/defect of Progress' reposition implementation:  
// if we are fetching on reposition, we have to load this last record into the  
// associated buffer(s), to truly emulate the GET NEXT semantic.  
last();  
afterReposition(false);
```

I believe this code was driven by a quirk I found with the AOG browse feature in the regression test environment (po.aog from the function browser). The quirky behavior was that the first time it was viewed, that browse would scroll the last record to the top of the browse, so it was the only one showing, and everything below it was blank (not a state you could reproduce by scrolling the browse by hand).

Perhaps this was not the correct implementation to reproduce that behavior, but I suspect it will break with this removed.

I guess the other changes are OK.

#### **#83 - 03/03/2015 03:47 AM - Stanislav Lomany**

Eric, in this file there is also updated history entry which says

Removed duplicate functionality in repositionByID (it is handled by Cursor)

This quirk is handled at Cursor:297 because this behaviour is shared with compound queries. So, removed code AdaptiveQuery was trying to make the same thing twice.

#### **#84 - 03/03/2015 09:31 AM - Eric Faulhaber**

Cursor:297 looks a little different to me. Have you confirmed the described behavior with the AOG Browse has not been lost? If so, please go ahead and commit the update.

#### **#85 - 03/03/2015 09:50 AM - Stanislav Lomany**

AOG Browse fails. Reverting the code in AdaptiveQuery doesn't fix the issue. I'll look into that.

**#86 - 03/03/2015 01:23 PM - Stanislav Lomany**

Eric, are you sure that scrolling to the last row (and making it the scroll row) is not a correct behavior?

**#87 - 03/03/2015 01:28 PM - Eric Faulhaber**

Correct or not, it's what Progress 9.1 did with the AOG Browse screen. It was quite strange. I've never recreated it with any other test case, but you may be able to model one after this code to test whether it still does that with later versions.

**#88 - 03/03/2015 02:25 PM - Stanislav Lomany**

OK, my bad. I misread your post. The last record to the top of the browse IS CORRECT. In this case my update works propely. Since there were no recent updates that required regression testing or merging, I'll check it in.

**#89 - 03/03/2015 02:50 PM - Stanislav Lomany**

svl\_upd20150302a was committed to bzs revision 10792.

**#90 - 03/06/2015 02:59 AM - Stanislav Lomany**

- File svl\_upd20150305a.zip added

Update for review. Fixes minor issues.

**#91 - 03/06/2015 08:46 AM - Greg Shah**

Code Review svl\_upd20150305a.zip

I'm OK with the changes.

Eric: please review the QueryWrapper change.

**#92 - 03/06/2015 02:40 PM - Stanislav Lomany**

One regression test has failed: P2JQuery.bufferHandle() returns invalid handle. Looking into it.

**#93 - 03/10/2015 05:50 AM - Stanislav Lomany**

About the recet bug. Here is the code that repositions browse to an invalid id:

```
ErrorHandler.silentErrorEnable();
query0.repositionByID(startRecid);
ErrorHandler.silentErrorDisable();
```

This code raises "cannot reposition" error which is supressed and turned into pending error. QueryWrapper.bufferHandle runs within silentErrorEnable/Disable bracket and returns handle by executing new handle(buf)

However handle.assign doesn't make assignment if there is a pedning error:

```
public void assign(WrappedResource value, boolean force)
{
    if (force || !ErrorHandler.isPendingError())
    {
        set(value);
    }
}
```

I assume that this check for pending errors was made for statements like

```
def var h as handle.  
h = .... no-error.
```

so I think we can use force handle assignment in this case:

```
handle h = new handle();  
h.assign(buf, true);  
return h;
```

**#94 - 03/10/2015 05:53 AM - Stanislav Lomany**

- File svl\_upd20150310a.zip added

Update for review. Relatively to the previous version of the update (0305a) only QueryWrapper has changed.

**#95 - 03/10/2015 09:59 AM - Greg Shah**

Code Review svl\_upd20150310a.zip

I'm fine with the changes.

Eric: please review the QueryWrapper change.

Before regression testing, you will also need to do a merge with bzt rev 10801.

**#96 - 03/10/2015 11:20 AM - Eric Faulhaber**

The QueryWrapper changes in 0310a look OK to me.

**#97 - 03/11/2015 10:06 AM - Stanislav Lomany**

- File svl\_upd20150311a.zip added

Update merged with the latest sources. Committed to bzt rev 10804.

**#98 - 03/13/2015 11:39 AM - Stanislav Lomany**

Guys, about the SCREEN-VALUE of a browse column. This attribute matches the content of the cell of the specific column in the current row. Browse column also has MODIFIED attribute which is true for the cells in the current row which were modified by editing on the current iteration.

SCREEN-VALUE state is not transferred in ScreenBuffer for two reasons:

- 1. It is not included in the list of the widgets belonging to a frame. My solution:  
1A. Store references to the BrowseColumn s in the Browse (currently these references are one way BrowseColumn -> Browse.)  
1B. Modify WidgetRegistry.getIds(int frameId) to include these column in the set of frame widgets.

- 2. Currently BrowseColumn instances permanently reside in UNINITIALIZED state. The correct state (for now) is probably UNCHANGED, but I'm confused on where should I set this state?

What do you think?

**#99 - 03/13/2015 12:06 PM - Greg Shah**

Treating the browse column as a real widget is going to add quite a bit of complication. It will have to be ignored all over the place on both the client and the server.

I think the browse column's screen-value and modified attribute is just the same as the currently focused cell's widget, right? So if we track that widget on the client side and allow the server to access that widget ID to find the current cell, then we can easily just read these values from the screen-buffer. This is much simpler and is less intrusive.

**#100 - 03/13/2015 12:22 PM - Stanislav Lomany**

Yes, but we have single cell's widget (there are multiple columns) and only for editable browse.

Also we can track these changes not involving screen buffer by tracking these changes in Browse/BrowseColumns and then passing them to BrowseWidget/BrowseColumnWidget by some in-browse communication.

**#101 - 03/16/2015 01:15 PM - Greg Shah**

Also we can track these changes not involving screen buffer by tracking these changes in Browse/BrowseColumns and then passing them to BrowseWidget/BrowseColumnWidget by some in-browse communication.

My concern with this is the following:

1. For all other widgets, SCREEN-VALUE support on the server is directly coupled to getting an updated screen buffer sent from the client.
2. We already have extensive support in the client and server for all the times that need screen buffers to be sent in either direction.

I don't want to add a separate mechanism for browse that then has to be called in all those same places. Please consider how we can add the browse data needed to the screen buffer, when it is needed.

And please also consider how to minimize change on that class' current functionality, data and performance.

**#102 - 03/16/2015 02:32 PM - Stanislav Lomany**

Are you planning to transfer browse columns states/values in screen buffers, like for any other widgets (e.g. fill-ins)?

**#103 - 03/16/2015 02:53 PM - Greg Shah**

I'm OK storing the state in the screen buffer but separate from other widgets. But I'm looking for a specific proposal from you. I don't want to create a



completely separate infrastructure for something as intrusive as screen-value support. I also don't want extra round trips to the server to send current state for browse, when this is the exact idea of the screen buffer.

#### #104 - 03/18/2015 10:51 AM - Stanislav Lomany

In order to get screen vales I've added the code which adds column values into a screen buffer (ThinClient.getScreenBuffer):

```
else if (comp instanceof Browse)
{
    Browse browse = (Browse) comp;
    int[] ids = browse.getColumnWidgets();

    for (int columId : ids)
    {
        BrowseColumn col = (BrowseColumn) registry().getComponent(columId);
        buffer.putScreenValue(columId, col.getValue());
    }
}
```

Also there is the code in GenericFrame.getScreenValue which allows "uninitialized" column widgets to pass through:

```
// quick out for uninitialized widgets
if (!(widget instanceof BrowseColumnWidget) && frameBuf.isUninitialized(wid))
{
    // always empty string
    return new character("");
}
```

In order to set screen value we have to properly determie data type in GenericFrame.setScreenValue:

```
// honor the type of the widget's data
Class<?> type = ((ControlEntity<?>) widget).getDataClass();

// convert the data to the right type if needed
if (!type.equals(character.class))
{
    if (type.equals(datetimetz.class))
    {
        val = new datetimetz(value);
    }
    else if (type.equals(datetime.class))
    {
        val = new datetime(value);
    }
    ...
}
```

For this we have to create special handing for browse columns OR make BrowseColumnWidget extend ControlEntity instead of GenericWidget. What do you think about all this changes?

**#105 - 03/18/2015 04:20 PM - Greg Shah**

In order to get screen vales I've added the code which adds column values into a screen buffer (ThinClient.getScreenBuffer):  
[...]

Do all of our client-side browse column widgets have a server-side BrowseColumnWidget associated with the same ID?

And does the server-side frame know about each of these BrowseColumnWidget instances?

Also there is the code in GenericFrame.getScreenValue which allows "uninitialized" column widgets to pass through:  
[...]

What does the 4GL return for an uninitialized BrowseColumnWidget? Is it different from a normal widget?

In order to set screen value we have to properly determie data type in GenericFrame.setScreenValue:  
[...]

For this we have to create special handing for browse columns OR make BrowseColumnWidget extend ControlEntity instead of GenericWidget.

If we don't have to do this in too many places, I think it is fine to put in custom logic (in this case, to call BrowseColumnWidget.getDataType() for browse columns).

**#106 - 03/18/2015 05:59 PM - Stanislav Lomany**

Do all of our client-side browse column widgets have a server-side BrowseColumnWidget associated with the same ID?  
And does the server-side frame know about each of these BrowseColumnWidget instances?

Yes in both cases.

What does the 4GL return for an uninitialized BrowseColumnWidget? Is it different from a normal widget?

In P2J browse columns are permanently in ScreenBuffer.UNINITIALIZED state so I had to ignore its state in order to process column as a normal widget.

**#107 - 03/18/2015 08:46 PM - Greg Shah**

OK, I'm fine with the current proposal.

**#108 - 03/20/2015 06:46 PM - Stanislav Lomany**

There is a quirk specific to single-selection browse in ChUI: if the current row cannot be fetched, it cannot be deleted from browse using any of DELETE-\* functions. The following error message is displayed: Invalid use of browse method DELETE-SELECTED-ROWS. There are no selected rows. Should I reproduce it? IMO it just reduces usability.

**#109 - 03/22/2015 03:59 PM - Stanislav Lomany**

Addition to the previous note. For multiple selection (ChUI and GUI) the message ("... There are no selected rows") is displayed if all of the deleted row(s) were physically deleted and THEN selected. By "physically" I mean that a record of those that compose the row was deleted. If the row is selected and then physically deleted, it can be deleted from browse. Unless you object, I'm not going to implement this feature.

**#110 - 03/23/2015 09:05 AM - Greg Shah**

We can't just let all these quirks wait until later. Both issues (in 108 and 109) seem like somewhat normal behavior that should be implemented now.

IMO it just reduces usability.

Not being compatible will cause us extra work later. For things that are pretty normal, we need to implement it in a compatible way from the beginning to avoid extra debugging and fixing later.

**#111 - 03/26/2015 09:43 AM - Stanislav Lomany**

A bug not related to browse which affect reproduction of the "... There are no selected rows" quirk.

Testcase:

```
def var v1 as char.

display v1 with frame fr.
enable v1 with frame fr.

on "x" anywhere do:
    message "update" update dummy as logical.
    pause message "pause".
end.

wait-for close of current-window.
```

Reproduction: x, enter.

At this point there is no cursor visible in 4GL and we can assume that v1 didn't receive the focus. In P2J cursor is blinking at v1. So, it looks like the focus is adjusted in P2J when MESSAGE ... UPDATE finishes, while in 4GL - when the trigger finishes.

**#112 - 03/31/2015 07:46 PM - Stanislav Lomany**

- File svl\_upd20150401a.zip added

Update for review. Added DELETE-\* functions, SCREEN-VALUE for browse columns and ROW-LEAVE triggers.

**#113 - 04/01/2015 10:21 AM - Greg Shah**

Code Review svl\_upd20150401a.zip

Constantin: please review the use of ThinClient.invokeTriggers() in Browse in this update.

Eric: please review the query changes.

1. BrowseWidget.deleteSelectedRow(integer) and BrowseWidget.deleteCurrentRow() may need to be optimized to only have 1 client side round trip instead of 3-4 round trips. The single trip down would have to return a code that would allow the proper error handling to be done on the server side. What is the feasibility of this?

2. BrowseWidget.deleteSelectedRows() seems like it will perform very poorly, with so many trips to the client. The optimization may be more important than with BrowseWidget.deleteSelectedRow(integer) but I haven't done the review of all the called code to see what server-side logic/state dependencies there may be.

3. I'm trying to think through the implications of directly invoking triggers from within the client-side browse. On the one hand, I like that the processing is very localized to the browse implementation. On the other hand, I worry that we will have to add features over time that we already have implemented in ThinClient and which would be utilized if we posted the ROW-LEAVE event and let the normal processing occur.

**#114 - 04/01/2015 11:28 AM - Stanislav Lomany**

1. BrowseWidget.deleteSelectedRow(integer) and BrowseWidget.deleteCurrentRow() may need to be optimized to only have 1 client side round trip instead of 3-4 round trips. The single trip down would have to return a code that would allow the proper error handling to be done on the server side. What is the feasibility of this?

I can reduce it to 2 trips - one for error check and one for refresh after the row has been deleted from the query result list.

**#115 - 04/01/2015 02:07 PM - Eric Faulhaber**

Greg Shah wrote:

Eric: please review the query changes.

The query changes look fine.

**#116 - 04/02/2015 05:57 AM - Constantin Asofiei**

Greg Shah wrote:

Constantin: please review the use of `ThinClient.invokeTriggers()` in Browse in this update.

I'm OK with the change, as all callpaths originate from the event processing.

**#117 - 04/03/2015 10:29 AM - Stanislav Lomany**

- File `svl_upd20150403a.zip` added

Update for review. DELETE-\* functions has been optimized, refresh after deletion of multiple rows has been implemented more precisely.

**#118 - 04/03/2015 12:55 PM - Greg Shah**

Code Review `svl_upd20150403a.zip`

I'm good with the changes.

**#119 - 04/08/2015 02:55 AM - Stanislav Lomany**

- File `svl_upd20150407a.zip` added

Committed to bzt rev 10831.

**#120 - 04/13/2015 06:15 AM - Stanislav Lomany**

COLUMN-BGCOLOR attribute is applicable to GUI only.

**#121 - 04/13/2015 12:18 PM - Stanislav Lomany**

Guys, the CURSOR-OFFSET and AUTO-ZAP attributes reflect corresponding attributes of the fill-in which represents the current cell. What do you think is more correct - write custom getters/setters OR create appropriate fields in `BrowseColumnConfig` and update then when fill-in values change?

**#122 - 04/13/2015 12:36 PM - Greg Shah**

In the 4GL many browse column attributes/methods are really just "redirected" from the contained control. I really don't want to duplicate all of that functionality, but as far as I understand we don't have a matching widget (e.g. fill-in) except possibly for the current cell.

In a perfect world, we would just unwrap the contained widget and read/write/execute the attribute/method on that instance.

write custom getters/setters

Do you mean that these new methods would call down to the client to set/get values?

create appropriate fields in `BrowseColumnConfig` and update then when fill-in values change

I thought we didn't have a contained fill-in for all cells in a browse. How would this work without them?

Also: Hynek is currently reworking our config syncing in a significant way. The results can be seen in [#2252](#) and may be ready this week sometime.

**#123 - 04/13/2015 01:55 PM - Stanislav Lomany**

My vision of two options explained:

Option 1 getter:

BrowseWidget.getSome --LogicalTerminal API call--> Browse.getSome --> FillIn.getSome

Option 1 setter:

BrowseWidget.setSome --LogicalTerminal API call--> Browse.setSome --> FillIn.setSome

Option 2 getter:

FillIn state updated --> BrowseColumn.config.some

BrowseColumn.config.some --TriggerProcessing pushes screen definitions--> BrowseColumnWidget.config.some

BrowseWidget.getSome --gets--> BrowseColumnWidget.config.some

Option 2 setter:

BrowseWidget.setSome --sets--> BrowseColumnWidget.config.some --push screen definitions--> BrowseColumn.config.some --sync with widget-->

FillIn.setSome

**#124 - 04/14/2015 08:33 AM - Greg Shah**

Option 1 is not feasible. The performance cost is too great. Attributes and methods can be called in loops. We can't embed a client trip in all of them, especially the getters. And there is a huge amount of extra code needed to redirect all of this to the client, which we don't want.

In the Option 2 setter, I don't understand what this means: --TriggerProcessing pushes screen definitions-->.

Is there an Option 3 where we actually maintain the current row of widgets (a widget for each browse column) and then delegate the state management to those widgets?

**#125 - 04/14/2015 09:48 AM - Stanislav Lomany**

In the Option 2 getter, I don't understand what this means: --TriggerProcessing pushes screen definitions-->.

That's a typo, I meant when a screen buffer is transferred to the server, for example during trigger processing.

Is there an Option 3 where we actually maintain the current row of widgets (a widget for each browse column) and then delegate the state management to those widgets?

Well, we can do that, but the problems are: 1. Maintaining these widgets is a time-consuming task. 2. We have to ensure that fill-in states are added to the screen buffer (even if they are invisible). 3. Potentially there may be cases that fill-in state may differ from browse column state, which most likely means adding browse-specific code into fill-in.

**#126 - 04/14/2015 09:52 AM - Eric Faulhaber**

Stanislav Lomany wrote:

3. Potentially there may be cases that fill-in state may differ from browse column state, which most likely means adding browse-specific code into fill-in.

Is it feasible to subclass here instead?

**#127 - 04/14/2015 09:57 AM - Stanislav Lomany**

Is it feasible to subclass here instead?

Yes, we can do that.

**#128 - 04/14/2015 05:12 PM - Stanislav Lomany**

- File `svl_upd20150415a.zip` added

Update for review. Contains SELECT-NEXT/PREV-ROW, MAX-DATA-GUESS and ROW-MARKERS.

**#129 - 04/15/2015 07:32 AM - Stanislav Lomany**

I'll put here rules which are applied to MAX-DATA-GUESS:

- Its default value is 100.
- For preselect queries it equals result size on query open
- For adaptive queries it is equals result size on query open if result size is less than vieport size or viewport size is more than 100 rows. Otherwise default value is used. Default value is changed by an actual value if we have reached the end or the number of actual rows exceeded 100.
- Row deletion decreases data guess, unless we use the default value.
- We can set data guess value, it remains active until we have scrolled forward or row has been deleted AND we have reached the end or the number of actual row is more than the assigned value.

**#130 - 04/15/2015 12:21 PM - Greg Shah**

Code Review `svl_upd20150415a.zip`

I'm good with the changes. Please get them regression tested.

**#131 - 04/15/2015 12:31 PM - Greg Shah**

Eric Faulhaber wrote:

Stanislav Lomany wrote:

3. Potentially there may be cases that fill-in state may differ from browse column state, which most likely means adding browse-specific code into fill-in.

Is it feasible to subclass here instead?

Does this idea make it more feasible to create an Option 3?

What might that look like?

**#132 - 04/15/2015 06:56 PM - Stanislav Lomany**

About REFRESHABLE. Typical use case of it is the following:

1. Set it OFF.
2. Make some repositions/re-open query, i.e. non-interactive stuff.
3. Set it ON and watch screen refreshed.

However nothing stops us from doing interactive actions on step 2, e.g. use navigation keys. Case of repositions intermixed with interactive actions are handled by 4GL surprisingly straightforward. But the case of query re-opening intermixed with interactive actions is a problem. Here is the brief set of rules for it:

1. After a query has been re-opened you are locked in the current view and can't go out of it unless:
2. If browse loses focus and re-gains it, we are scrolled to the first row of the OLD query result set (this query is supposed to be closed).
3. HOME and END buttons bring us to the first or last pages of the new query and makes browse fully usable for interactive actions.

GUI behavior differs in details, e.g. it doesn't have option [#2](#).

I don't plan to implement option [#2](#) and working to determine the full set of rules for query re-opening.

**#133 - 04/15/2015 08:34 PM - Greg Shah**



The plan is fine. How long do you need for the implementation?

**#134 - 04/16/2015 08:18 PM - Stanislav Lomany**

The set of rules for REFRESHABLE. By the "repositioned row" I mean the last row the query was repositioned to.

### **Repositions only case.**

After REFRESHABLE becomes ON:

1. For read-only single selection browse: if there were repositions - we scroll to the current row. If the repositioned row is in the new view - it becomes the current row.
2. For read-only multiple selection browse: if there were repositions - we scroll to the last selected row, if any. If the repositioned row is in the new view - it becomes the current row. Otherwise the current viewport row is preserved. If there are no selected rows, we scroll to the repositioned row
3. For editable browse: scroll to the repositioned row.

### **Opening and optional repositions case.**

After a query has been re-opened we are in the "weird" mode:

1. We are locked in the current view and can't go out of it.
2. In this mode we have fake selection. We can select rows, but that doesn't count as actual selection when it comes to row management.
3. If you try to go out of bounds, you will stuck in this mode even if REFRESHABLE is set to yes (yes, that makes browse completely unusable).
4. If browse loses focus and re-gains it, we are scrolled to the first row of the OLD query result set (this query is supposed to be closed).
5. More quirks for multiple and editable modes.
6. More quirks for GUI.
7. However HOME and END buttons brings us to the first or last pages of the new query and makes browse fully usable for interactive actions (regardless of REFRESHABLE state).

Additional aspects:

1. Selection is cleared before entering the "weird" mode.
2. After REFRESHABLE becomes ON: if there are no repositions - we scroll to the first row preserving viewport row, if the repositioned row is in the view - it becomes the current row. If there were repositions - scroll to repositioned row.

I suggest to NOT implement the "weird" mode (1-7). A lot of problems and no evidence that a user will observe it in a real converted application. Implementation of other aspects ("repositions only case" and "additional aspects") should take less than a day.

**#135 - 04/17/2015 03:56 AM - Stanislav Lomany**

- File `svl_upd20150416a.zip` added

Implemented SELECT-NEXT/PREV-ROW, MAX-DATA-GUESS and ROW-MARKERS. Committed to bzt rev 10839.

**#136 - 04/17/2015 09:37 AM - Greg Shah**

I suggest to NOT implement the "weird" mode (1-7).

Agreed.

**#137 - 04/22/2015 07:25 AM - Stanislav Lomany**

Conversion bug - two temp-tables with different labels / column labels are converted to the same TempRecord class. Example: test.p:

```
def temp-table ttl
    field f1 as integer label "label 1".
create ttl.
```

test2.p:

```
def temp-table ttl
    field f1 as integer label "label 2".
create ttl.
```

Because this is not a browse problem and I cannot fix it quickly, I'll return to it after all other browse work is done.

**#138 - 04/22/2015 09:55 AM - Greg Shah**

Because this is not a browse problem and I cannot fix it quickly, I'll return to it after all other browse work is done.

Please create a bug task in the Database project. Let Eric deal with it. You don't need to fix this.

**#139 - 04/29/2015 01:13 AM - Stanislav Lomany**

I need to create a browse column.

1. I assumed that it should be a dynamic widget. Is that correct? I've created it using `GenericWidget(boolean dynamic = true, T config)` c'tor and assigned id generated by `WidgetId.nextID().asInt()`.
2. Should I worry about its cleanup? If yes, then how can I delete it?
3. After Hynek's update it just stopped working without any errors. It turned out that some variables were not properly initialized and I made the explicit call `columnWidget.afterConfigUpdate(columnWidget.config)` on the server side to fix it.

**#140 - 04/29/2015 10:10 AM - Greg Shah**

I assumed that it should be a dynamic widget. Is that correct?

I doubt it. Unfortunately, the 4GL docs suggest that the DYNAMIC attribute only can be read from a browse and not a browse column. So it may be hard to tell.

But dynamic widgets have very different behavior (e.g. scoping/lifetime) than do static widgets. It doesn't seem like a static browse should have automatically generated dynamic column widgets.

Should I worry about its cleanup? If yes, then how can I delete it?

If it is not dynamic, this isn't a problem. If it must be dynamic, then it definitely needs cleanup but that would require special cleanup logic when the static browse permanently goes out of scope.

After Hynek's update it just stopped working without any errors. It turned out that some variables were not properly initialized and I made the explicit call `columnWidget.afterConfigUpdate(columnWidget.config)` on the server side to fix it.

I have no ideas here.

**#141 - 04/30/2015 02:17 PM - Greg Shah**

I've added the GUI-specific options, attributes and methods to the [#2564](#) task (creation of the GUI BROWSE widget).

Am I correct in understanding that the following attributes are not currently implemented and won't be in the next update:

NEXT-COLUMN  
MIN-HEIGHT-CHARS  
SORT  
CURRENT-ROW-MODIFIED  
REFRESHABLE  
NEW-ROW  
CREATE-ON-ADD (undocumented but found in the customer GUI project)

Method which is not implemented yet: GET-REPOSITIONED-ROW

If I understand correctly, then we have the following categories:

1. features currently being implemented
2. attributes/method listed above as still needed
3. GUI options/methods/attributes moved to [#2564](#)
4. options that were dropped because we think they have no known behavior

Please confirm if everything else is already implemented.

**#142 - 04/30/2015 02:20 PM - Greg Shah**

I've checked the GET-BROWSE-COLUMN method and although it is implemented, I see that it is supposed to return a HANDLE but today we return a BrowseColumnWidget instance.

I think we need to fix that. What is the breakage involved?

**#143 - 05/01/2015 05:04 AM - Stanislav Lomany**

Here is the list of all remaining features:

**GUI-specific**  
FIT-LAST-COLUMN/EXPANDABLE  
NO-COLUMN-SCROLLING  
ROW-HEIGHT-CHARS  
COLUMN-BGCOLOR  
MOVE-COLUMN  
COLUMN-MOVABLE  
COLUMN-RESIZABLE  
ALLOW-COLUMN-SEARCHING  
MIN-HEIGHT-CHARS  
SORT, MAX-CHARS - I think that actually means implementation of drop-down cells  
ROW-HEIGHT-CHARS  
LABEL-BGCOLOR  
SCROLLBAR-VERTICAL

**No known behavior**

OVERLAY  
CENTERED  
SCROLLABLE

**Expected to have implemented**

INSERT-ROW  
CURRENT-ROW-MODIFIED  
NEW-ROW  
CREATE-ON-ADD  
GET-BROWSE-COLUMN  
AUTO-ZAP  
CURSOR-OFFSET

**Missed it, try to fit in time**

GET-REPOSITIONED-ROW

**#144 - 05/01/2015 05:06 AM - Stanislav Lomany**

If I understand correctly, then we have the following categories:

1. features currently being implemented

Kind of. As far as I've seen, pretty much every aspect of browse works a bit differently in GUI.

2. attributes/method listed above as still needed
3. GUI options/methods/attributes moved to [#2564](#)
4. options that were dropped because we think they have no known behavior

**#145 - 05/03/2015 12:32 PM - Stanislav Lomany**

- File `svl_upd20150503a.zip` added

Update for review - REFRESHABLE and ADD-LIKE-COLUMN.

**#146 - 05/03/2015 01:23 PM - Greg Shah**

Code Review svl\_upd20150503a.zip

Eric: please review.

1. Shouldn't there be a methods\_attributes.rules change to correspond with the buffer handle attribute?
2. For ThinClient.executePendingRefresh() and ThinClient.notifyColumnsUpdated(), can we check for null before executing the eventDrawingBracket()? When the browse is null, does the empty bracket execution do something useful?
3. Browse.updateLabelsHeigth() should probably be renamed Browse.updateLabelsHeight().
4. Are there any conversion changes needed to support the getBrowseColumn() now returning a handle?

**#147 - 05/04/2015 03:19 AM - Stanislav Lomany**

- File svl\_upd20150504a.zip added

All the flaws fixed.

**#148 - 05/04/2015 09:35 AM - Greg Shah**

Code Review svl\_upd20150504a.zip

1. In ThinClient.queryClosed() and ThinClient.refreshScrollRow(), the call to clearPutScreenInMessageArea() should also be inside the null check.
2. I think your choice to move GET-BROWSE-COLUMN() to the BrowseInterface is a good one. That means that it should be removed from CommonWidget and GenericWidget.

**#149 - 05/04/2015 11:01 AM - Eric Faulhaber**

20150504a looks OK to me from the persistence side.

**#150 - 05/05/2015 09:52 PM - Stanislav Lomany**

About CREATE-ON-ADD. At this point it cannot be implemented because:

1. I do not quite understand what it does. Here is the description from some documentation:

Default behavior is different for Progress and non-Progress databases. For Progress databases, the default is NO. For non-Progress databases and temp-tables, the default is YES. This is because initial values are displayed using the "template" record from the schema, which is usable only for Progress database tables.

If the Create-On-Add attribute is set to YES, then the new database record is created and saved when the add-record event is dispatched (for example, when the end user chooses the Add button in an Update SmartPanel).

If the value of Create-On-Add is NO, then the new record is not created until the update-record event is dispatched (for example, when the end user chooses the "Save" button). When Create-On-Add is set to NO, initial values will be displayed automatically only for Progress database tables. For non-Progress databases and temp-tables, initial values must be displayed by application code.

From the observations, when it is turned on, it does something strange instead - updates existing records instead of creating a new one. 2. Even if I get how it works, there are unimplemented features that are required to support full record creation cycle in browse in which CREATE-ON-ADD participates:

- ASSIGN buf.
- brws:CREATE-RESULT-LIST-ENTRY.

So CREATE-ON-ADD is added to the GUI list.

#### #151 - 05/06/2015 09:35 AM - Greg Shah

So CREATE-ON-ADD is added to the GUI list.

OK, please add notes to [#2564](#) so that this is not missed.

Also: if you have any questions or are looking for some ideas, please feel free to contact Guy Mills. He is an expert level 4GL developer including strong knowledge of SmartObjects. He may be able to point you in the right direction.

The best way to contact him is to add him as a watcher to a given task and then post the questions there. That way the discussion is captured.

#### #152 - 05/06/2015 12:08 PM - Stanislav Lomany

- File svl\_upd20150505a.zip added

Implemented REFRESHABLE and ADD-LIKE-COLUMN. Committed to bzt rev 10848.

#### #153 - 05/08/2015 07:24 AM - Stanislav Lomany

I have some things about AUTO-ZAP/CURSOR-OFFSET/CURRENT-ROW-MODIFIED to clear:

1. How do we expect to implement attributes like this? As we have discussed above (note 124) we can have a set of fill-ins, one for each browse column. What should it be in this case - set of FillIns and corresponding FillInWidgets on the server side with transparent data transmission between them?
2. Should they be static or dynamic? I can add a dynamic widget to frame, but that changes frame layout reserving place for them. I don't know how to add a static widget to frame - maybe register it manually on the client side?
3. How FillIn should push data to its FillInWidget counterpart? AUTO-ZAP is a simple config.autoZap variable, but FillInWidget.isAutoZap() doesn't seem to work.
4. CURSOR-OFFSET is non-implemented feature, but it has implementation on the client side:

```
public int getCursorOffset()
{
    return (config.appDataPres != null) ? config.appDataPres.getCursorPos() : offsCursor;
}
```

How data transmission should happen in this case?

What should it be in this case - set of FillIns and corresponding FillInWidgets on the server side with transparent data transmission between them?

This is a good question.

On the server side for this to be useful, we would certainly need FillInWidget instances (or ComboBoxWidget...) for each column.

What I don't know is if we really need new client-side widgets to match. Don't we already have a client-side widget for each column?

I guess I prefer not to add new widgets unless we really need to do so. I wonder if we can just take the FillInConfig state and pushed it down in some easier way.

Should they be static or dynamic? I can add a dynamic widget to frame, but that changes frame layout reserving place for them. I don't know how to add a static widget to frame - maybe register it manually on the client side?

They should not exist in the frame itself. These are a hidden attribute of the browse column itself and definitely should not be considered frame-level widgets.

How FillIn should push data to its FillInWidget counterpart? AUTO-ZAP is a simple config.autoZap variable,

This is the question I'm wondering about too. Does this state transfer need to be bidirectional?

but FillInWidget.isAutoZap() doesn't seem to work.

I'm not sure what you mean by this.

CURSOR-OFFSET is non-implemented feature, but it has implementation on the client side:

...

How data transmission should happen in this case?

I'm not sure. I can say this:

1. We want to minimize duplication of functionality between browse columns and the real widget being used.
2. We want to minimize the duplication of APIs and delegation of calls. For example, rather than add APIs for methods/attributes to the BrowseColumnWidget, I prefer to unwrap the contained widget (e.g. FillInWidget) and use the already existing APIs there.
3. We want to avoid unnecessary client/server round trips. Where we can just let the state sync normally and calculate or report results on the other side as needed, that is best.
4. We don't want to complicate layout, frame processing and so forth by creating widgets in a way that will require code all over the project to be aware of this "special case".
5. I prefer not to make the screen-buffer processing over complicated.



**#155 - 05/08/2015 12:15 PM - Greg Shah**

Some planning info from SVL:

Here are the items left:

1. INSERT-ROW basics - Will send today
2. AUTO-ZAP/CURSOR-OFFSET/CURRENT-ROW-MODIFIED transport - That depends on the results of our discussion, but I think 1 day for usual solution, 2 days for complicated.
3. MODIFIED fill-in
4. AUTO-ZAP fill-in (details later)
5. CURSOR-OFFSET fill-in
6. GET-REPOSITIONED-ROW - Seems simple, 0.5 - 1 day.
7. Thorough implementation of INSERT-ROW - 1 day for browse part. However may contain fill-in problems.

Regarding 3, 4, 5 - should I solve fill-in problems? 3 and 4 may potentially turn out to be complicated.

So without fill-in problems it is 3-4 days.

**#156 - 05/08/2015 12:19 PM - Greg Shah**

Regarding 3, 4, 5 - should I solve fill-in problems? 3 and 4 may potentially turn out to be complicated.

Yes, let's get this done properly the first time.

**#157 - 05/08/2015 01:00 PM - Stanislav Lomany**

Don't we already have a client-side widget for each column?

We have single fill-in which is created and then destroyed each time when we switch to the next cell.

Does this [AUTO-ZAP] state transfer need to be bidirectional?

Yes.

4GL and P2J behave differently:

1. Zap state equals AUTO-ZAP value on fill-in create.
2. After user input zap state may be reset, and in 4GL AUTO-ZAP returns actual zap state, while in P2J - initial AUTO-ZAP value.

3. In 4GL you can set AUTO-ZAP to turn it on again during editing.

**#158 - 05/08/2015 01:04 PM - Stanislav Lomany**

P2J has pushScreenDefinition() to sync server config with client. Do we have anything that makes similar action in reverse direction?

**#159 - 05/08/2015 01:48 PM - Greg Shah**

P2J has pushScreenDefinition() to sync server config with client. Do we have anything that makes similar action in reverse direction?

There is no need for that. So long as the values in the config are modified, they will be synchronized to the server automatically the next time processing shifts to the server.

The only reason we have a pushScreenDefinition() is because setting attributes often have an immediate affect that can be visible to the end user. These must be implemented synchronously and cannot wait until the next natural trip to the client.

But the client state cannot be accessed on the server except when flow of control shifts there (through a trigger, validation expression or the return from some editing mode). At that time all config changes will be affected.

Do you have a need for something different?

**#160 - 05/08/2015 04:11 PM - Stanislav Lomany**

- File svl\_upd20150508a.zip added

Update for review. Contains INSERT-ROW/NEW-ROW basics. There are missing fixes for empty browse and "Cannot insert row, browse is read only", I'll add them a bit later.

**#161 - 05/08/2015 04:23 PM - Greg Shah**

Code Review svl\_upd20150508a.zip

It looks fine.

Please keep building the additional work for this week (2-6) on top of this update. In other words, don't regression test and check in this update, but instead aggregate the changes and do the testing/check-in later in the week. The reason is that we have several people trying to get things through testing right now and I don't want to overload devsrv01.

Your update will also need to be merged after they check in because there are conflicting files...

**#162 - 05/10/2015 01:59 AM - Stanislav Lomany**

So long as the values in the config are modified, they will be synchronized to the server automatically the next time processing shifts to the server. ... Do you have a need for something different?

As far as I see, we sync widget states (changed/unchanged), variable values, header values, screen values and not config values.

**#163 - 05/10/2015 02:05 AM - Constantin Asofiei**

Stanislav Lomany wrote:

So long as the values in the config are modified, they will be synchronized to the server automatically the next time processing shifts to the server. ... Do you have a need for something different?

As far as I see, we sync widget states (changed/unchanged), variable values, header values, screen values and not config values.

The configs are not sync'ed via the StateSynchronizer - to put it simply, they are automatically picked up by AspectJ and ConfigManager is notified whenever an assignment is done to a config field; so, this processing is somehow hidden and that's why is important to either use p2j.jar in your testing or ensure AspectJ is integrated properly with your IDE.

**#164 - 05/10/2015 02:44 AM - Stanislav Lomany**

Constantin, I change some FillIn.config.val on the client side, then a trigger is invoked and I get FillInWidget.config.val. It is not updated. Isn't that how it suppose to work? If yes, then what may be missing? I use p2j.jar for testing.

**#165 - 05/10/2015 04:05 AM - Constantin Asofiei**

Stanislav Lomany wrote:

Constantin, I change some FillIn.config.val on the client side, then a trigger is invoked and I get FillInWidget.config.val. It is not updated. Isn't that how it suppose to work? If yes, then what may be missing? I use p2j.jar for testing.

If the field is not transient then it should have been picked up by the ConfigManager as "dirty" and transfer it to the server.

Also, not all config field usage is tracked - if you are outside of the p2j.ui package, then AspectJ will not pick it up. Please check the annotation at ConfigFieldSetterAspect.beforeSetField for the cases where the config fields are tracked.

**#166 - 05/10/2015 09:44 PM - Stanislav Lomany**

Greg, about AUTO-ZAP: I think we can create FillInConfig.zapActive variable and update it each time zap state it is update.

About CURSOR-OFFSET: we can use unused variable FillInConfig.cursorOffset for that.

About data transmission:

1. Create a set of FillInWidgets on the server.
2. Send the set of FillInConfigs to the client.
3. Editable cell is created using one of configs on the client side.

**#167 - 05/11/2015 01:03 AM - Greg Shah**

I like the approach. It improves compatibility on the server-side while not requiring an explosion of new widgets (and lots of change) on the client.

Go with it.

**#168 - 05/11/2015 11:28 AM - Stanislav Lomany**

Until SET-REPOSITIONED-ROW is implemented, GET-REPOSITIONED-ROW always returns 1.

**#169 - 05/17/2015 11:07 AM - Stanislav Lomany**

I've implemented CURSOR-OFFSET for fill-in, but there is one issues, which I think deserves a separate task - 4GL and P2J has different cursor behavior for numeric values. In an "\_\_\_12345" field 4GL allows to move cursor in underscored area, while P2J doesn't allow that. I suppose that there should be changes in data presentation - NumberBuf considers the first digit as cursor offset = 1 and in 4GL the cursor offset = 1 when it matches the first position in fill-in.

**#170 - 05/17/2015 11:14 AM - Greg Shah**

I agree. Please open a separate task for this.

**#171 - 05/26/2015 01:23 PM - Stanislav Lomany**

CURRENT-ROW-MODIFIED logic differs from MODIFIED logic. The latter is tricky, while CURRENT-ROW-MODIFIED is straightforward: a row is "modified" if

1. the value of the current cell differs from the saved (on row entry) value
- OR
2. we left a cell with the value which was different from the saved value.

**#172 - 06/08/2015 05:11 PM - Stanislav Lomany**

- File svl\_upd20150608a.zip added

Update for review. INSERT-ROW, CURSOR-OFFSET, AUTO-ZAP and CURRENT-ROW-MODIFIED.

**#173 - 06/10/2015 04:52 AM - Stanislav Lomany**

- File svl\_upd20150610a.zip added

Removed excess function isNewRow from the previous update.

**#174 - 06/10/2015 10:49 AM - Greg Shah**

Code Review svl\_upd20150610a.zip

In regard to this comment // Oddly enough, 4GL returns zap state of the FOCUSED fill-in, not necessarily this one, please post the code that shows

this behavior.

Otherwise the code looks good. When do you expect to have everything 100% complete? Igor's work in [#2534](#) is dependent upon your changes for AUTO-ZAP and CURSOR-OFFSET in fill-in.

**#175 - 06/10/2015 11:36 AM - Stanislav Lomany**

When do you expect to have everything 100% complete? Igor's work in [#2534](#) is dependent upon your changes for AUTO-ZAP and CURSOR-OFFSET in fill-in.

AUTO-ZAP and CURSOR-OFFSET are done except separate task [#2576](#). Let me know if I should work on this task first.

**#176 - 06/10/2015 12:11 PM - Stanislav Lomany**

In regard to this comment // Oddly enough, 4GL returns zap state of the FOCUSED fill-in, not necessarily this one, please post the code that shows this behavior.

```
def var i as char init "abc".
def var j as char init "def".

display i j with frame f1.
enable i j with frame f1.

on "z" anywhere do:
    message string(i:auto-zap in frame f1).
end.

message "tab, z, z".
message "output should be yes, no".

wait-for close of current-window.
```

**#177 - 06/10/2015 06:13 PM - Greg Shah**

AUTO-ZAP and CURSOR-OFFSET are done

OK.

except separate task [#2576](#). Let me know if I should work on this task first.

No, you don't need to work that.

What else is open in this task?

**#178 - 06/11/2015 07:12 AM - Stanislav Lomany**

What else is open in this task?

My checklist:

1. Page up/down, home/end doesn't work in editable browse.
2. Uninitialized fill-in should be left in this state on row leave. Especially actual for numeric fields.
3. Check how row selection works for new rows.
4. Preserve current empty new row on refresh (other rows are cleared).
5. In 4GL (ChUI) last/first rows has invalid row value. Reproduce?
6. Check how addition of new row works in multiple selection mode (last selected row is used?).
7. brws-insert-row1.p: i, j, up - in P2J browse doesn't return to the first row.
8. Check if new/existing rows are undone or committed on insert-row() call.

**#179 - 06/11/2015 09:04 AM - Greg Shah**

I would like you to please get the current update through regression testing and checked in. This will free up other people whose work is dependent.

While testing is going, you can work on the checklist as a separate update.

In 4GL (ChUI) last/first rows has invalid row value. Reproduce?

Probably. Please show an example so I understand the idea.

#180 - 06/11/2015 09:21 AM - Stanislav Lomany

I would like you to please get the current update through regression testing and checked in. This will free up other people whose work is dependent.

I've fixed a regression, testing is re-running.

Probably. Please show an example so I understand the idea.

Rows 1 and 3 are new rows. Both were generated using INSERT-ROW. Row 3 is empty and that is how it supposed to be. Row 1 has duplicated value of the *current* column because it is situated before the first row (the same applies to the rows after the last row). That doesn't always happens, some additional conditions apply (to be investigated). ChUI only.

				<btn1>	
s label f2 f3					
*>	1,234				
*	1,234	10	11/06/15		
*					
*	2	20	11/06/15		
*	3	30	11/06/15		
*	4	40	11/06/15		

**#181 - 06/11/2015 09:31 AM - Greg Shah**

Yes, you should go ahead and implement it. Inserting before the first or after the last seems like something that is common enough.

**#182 - 06/14/2015 04:41 AM - Stanislav Lomany**

- File `svl_upd20150612a.zip` added

INSERT-ROW, CURSOR-OFFSET, AUTO-ZAP and CURRENT-ROW-MODIFIED. Committed to bzt rev 10879.

**#183 - 06/17/2015 11:37 AM - Greg Shah**

What else is open in this task?

My checklist:

...

Doesn't GET-REPOSITIONED-ROW still need to be done?

**#184 - 06/17/2015 04:55 PM - Stanislav Lomany**

Doesn't GET-REPOSITIONED-ROW still need to be done?

GET-REPOSITIONED-ROW always returns 1 unless the row was set by SET-REPOSITIONED-ROW which doesn't present in the original list. Let me know if I should implement SET-REPOSITIONED-ROW.

**#185 - 06/17/2015 05:05 PM - Greg Shah**

I don't understand. As far as I can see, we already support SET-REPOSITIONED-ROW both at runtime and in conversion:

BrowseWidget.java

```
/**
 * Implements the SET-REPOSITIONED-ROW() widget method.
 *
 * @param row
 *        1-based row number where the new record is displayed
 * @param cond
```



```

*         if true, tries to reposition to the browse
*         viewport existing row first
*
* @return  Always true.
*/
public logical setRepositionedRow(int row, boolean cond)
{
    LogicalTerminal.setRepositionedRow(this, row - 1, cond);
    return new logical(true);
}

/**
* Provides the SET-REPOSITIONED-ROW method which assigns the row number and type of
* repositioning for the browse.
*
* @param    row
*           1-based row number where the new record is displayed
* @param    cond
*           If this parameter is "conditional", then method tries to
*           reposition to the browse viewport existing row first
*
* @return    The result of operation. It evaluates to true if
*            operation was successful.
*/
public logical setRepositionedRow(int64 row, String cond)
{
    if (row == null || row.isUnknown())
    {
        return new logical(false);
    }

    return setRepositionedRow(row.intValue(), cond);
}

/**
* Provides the SET-REPOSITIONED-ROW() method which assigns the row number and type of
* repositioning for the browse.
*
* @param    row
*           1-based row number where the new record is displayed
* @param    cond
*           If this parameter is "conditional", then method tries to
*           reposition to the browse viewport existing row first
*
* @return    The result of operation. It evaluates to true if
*            operation was successful.
*/
public logical setRepositionedRow(int row, String cond)
{
    if (cond == null)
    {
        return new logical(false);
    }

    return setRepositionedRow(row, "conditional".startsWith(cond.toLowerCase()));
}

```

#### methods\_attributes.rules

```

<rule>list.put(prog.kw_set_rpos, execLib("cr_descr", "BrowseInterface" , "setRepositionedRow" ,
null , true ))</rule>

```

**#186 - 06/18/2015 02:30 AM - Stanislav Lomany**

It is not implemented at runtime (Browse.java):

```
public void setRepositionedRow(int n, boolean conditional)
{
    // TODO: implement it
}
```

**#187 - 06/23/2015 07:35 AM - Stanislav Lomany**

Greg,

Check if new/existing rows are undone or committed on insert-row() call.

Actually rows get to the quirky state. The scenario:

1. Update some row 1.
  2. Leave it by calling INSERT-ROW in a trigger. Row 1 will contain updated value.
  3. You can update the new row 2 if you want.
  4. If you return to the row 1 and then leave it (e.g. you can move back to row 2) then row 1 is committed.
  5. If you leave row 2 to any row except row 1 then value of row 1 will remain in cache but is NOT committed.
- Should I reproduce this quirk?

Side note: in multi-selection browse if a new row is selected and DELETE-CURRENT-ROW/DELETE-SELECTED-ROW was called, selection in 4GL changes to invalid (i.e. one wouldn't expect that these rows will be selected). I decided not to implement it.

**#188 - 06/23/2015 10:11 AM - Stanislav Lomany**

Addition to note 180 ("In 4GL (ChUI) last/first rows has invalid row value"). Unless it is the first column, you cannot change copied value (only navigation and trigger keys work) until you re-enter the cell. To what extent do we want to reproduce this item in the checklist?

**#189 - 06/23/2015 02:23 PM - Greg Shah**

Stanislav Lomany wrote:

It is not implemented at runtime (Browse.java):

[...]

This is not good. I did not include it on the list because based on my review of methods\_attributes.rules and BrowseWidget.java, it appeared to be implemented. Normally, if a method is implemented all the way out through LogicalTerminal, ThinClient and ClientExports, I don't check Browse.java to see if it actually exists. I didn't think we stubbed things out that far unless the code is really there. The customer code uses SET-REPOSITIONED-ROW 2056 times. It is definitely needed. But if I think things are already supported, then I don't add them to the list of work to be done...

Please make a list of browse features (methods, attributes, options) which are stubbed out but which have no runtime functionality implemented. I will have to double check that list to see if there are other things to be added.

It is also not clear to me how much GET-REPOSITIONED-ROW() is dependent upon the REPOSITION statement. We already support REPOSITION. The 4GL docs suggest these are linked. Even just the presence of GET-REPOSITIONED-ROW() suggests that we need a real implementation. In the future, please ask the question before assuming that we can hard-code it to 1 and postpone further work.

**#190 - 06/23/2015 02:27 PM - Greg Shah**

Actually rows get to the quirky state. The scenario:

...

Should I reproduce this quirk?

What does the implementation do today when there is an uncommitted update and INSERT-ROW() is called?

**#191 - 06/23/2015 02:29 PM - Greg Shah**

Side note: in multi-selection browse if a new row is selected and DELETE-CURRENT-ROW/DELETE-SELECTED-ROW was called, selection in 4GL changes to invalid (i.e. one wouldn't expect that these rows will be selected). I decided not to implement it.

This one seems easy to implement. It seems like it will be worse in the future to debug a problem and find out this is missing than to just implement it now. Is it harder than I am guessing?

**#192 - 06/23/2015 02:31 PM - Greg Shah**

Unless it is the first column, you cannot change copied value (only navigation and trigger keys work) until you re-enter the cell. To what extent do we want to reproduce this item in the checklist?

This part seems obscure. Don't do it.

**#193 - 06/23/2015 02:35 PM - Greg Shah**

It seems that there are a very large number of postponed browse quirks already. We need a much better approach to tracking those than just documenting them in this task.

Please do the following:

1. Create a new task in the UI project. Call it "implement missing browse quirks" and do not set a target version.
2. For each quirk that is unimplemented, please create a new task. Make that task a child task of the "implement missing browse quirks" task, so that we can look there to see a single list of all missing quirks.
3. Put enough detail in each quirk task to understand what has to be implemented.
4. Add the "implement missing browse quirks" task as a "related issue" in this task.

**#194 - 06/23/2015 03:28 PM - Stanislav Lomany**

What does the implementation do today when there is an uncommitted update and INSERT-ROW is called

At that time I made a safe bet: value is undone.

This one seems easy to implement. It seems like it will be worse in the future to debug a problem and find out this is missing than to just implement it now. Is it harder than I am guessing?

That happens in the following area:

1. We should have multi-selection editable browse.
2. We should switch to read-only mode (using REPLACE) and select multiple rows.
3. In the set of these rows there should be new uncommitted rows.
4. We should call DELETE-\* -ROW.

That is pretty rare, but very buggy case in terms of 1. in-browse position after deletion and 2. selection of remaining rows. By "very buggy" I mean time and effort-consuming. Also, from usability point I think it is dangerous to have something which alters selection outside the browse view as the user may perform operations with not that rows he intended to.

**#195 - 06/23/2015 03:42 PM - Greg Shah**

For both of these quirks, can you easily detect that the problem would occur? If so, then let's detect those cases (and only those cases) and when detected call `UnimplementedFeature.todo()` so that we get some indication in the log that it is needed. Make each message easy to track back to the matching quirk.

Then add the quirks to the list of unimplemented browse quirks.

**#196 - 06/23/2015 05:34 PM - Stanislav Lomany**

Please make a list of browse features (methods, attributes, options) which are stubbed out but which have no runtime functionality implemented.

I found  
SET-REPOSITIONED-ROW  
MODIFIED  
LABEL-BGCOLOR  
MOVE-COLUMN  
MIN-HEIGHT-CHARS

**#197 - 06/24/2015 10:53 AM - Greg Shah**

SET-REPOSITIONED-ROW - used  
MODIFIED - used  
LABEL-BGCOLOR - used  
MOVE-COLUMN - used  
MIN-HEIGHT-CHARS - not sure; it is referenced in an internal procedure but that procedure is not called from anywhere using a hard coded name, it is possible (but unlikely) that it is called through some dynamic mechanism

The first 4 are needed for sure. To be safe, please implement all 5.

Plan:

1. Please finish the work on the checklist items in note 178. Get those changes tested and checked in.
2. Implement the runtime support for the above methods and attributes (and GET-REPOSITIONED-ROW).

Does this make sense?

**#198 - 06/24/2015 10:59 AM - Stanislav Lomany**

Yes, but LABEL-BGCOLOR, MOVE-COLUMN and MIN-HEIGHT-CHARS are GUI-specific.

#199 - 06/24/2015 11:04 AM - Greg Shah

LABEL-BGCOLOR, MOVE-COLUMN and MIN-HEIGHT-CHARS are GUI-specific.

OK, I see they are already listed in [#2564](#). So step 2 will be to implement SET-REPOSITIONED-ROW, GET-REPOSITIONED-ROW and MODIFIED.

#200 - 06/24/2015 05:34 PM - Stanislav Lomany

Addition to "In 4GL (ChUI) last/first rows has invalid row value." behavior: if the value of the current cell was modified on the current iteration, then it is *moved* to the new row (i.e. the value is copied and is undone for the current row).

#201 - 06/24/2015 07:11 PM - Stanislav Lomany

Regarding "Check how addition of new row works in multiple selection mode (last selected row is used?)". Looks like another quirk to do later. To trigger it, one should add a new row in REPLACE mode.

Scenario:

f1 f2	
*	1 text10
*	2 text20
*	3 text30
*>	4 text40
*	5 text50
*	6 text60

After the new row has been added and the value was manually updated:

f1 f2	
*	1 text10
*	2 text20
*>	333 text30
*	4 text40
*	5 text50
*	6 text60

After scrolling out of the view and then back again:

--

	f1	f2
*>	1	text10
*	2	text20
*	3	text30
*	333	text30
*	4	text40
*	5	text50

So the new row was actually added, but the rows above it were not shifted properly.

#### #202 - 06/25/2015 07:43 AM - Greg Shah

OK, add this to the quirks list.

#### #203 - 06/25/2015 04:45 PM - Stanislav Lomany

- File `svl_upd20150625a.zip` added

Update for review. Fixes issue from the checklist in item 178.

#### #204 - 06/25/2015 06:38 PM - Greg Shah

Code Review `svl_upd20150625a.zip`

I'm OK with the changes. Please get them regression tested.

#### #205 - 06/30/2015 02:11 PM - Stanislav Lomany

The latest update was converted to branch 2422a. Committed to bazaar as P2J rev. 10892.

#### #206 - 07/01/2015 07:43 AM - Constantin Asofiei

Stanislav, is there a query-related reason why the browse columns (at the widget level) are linked with the `BrowseModel` (which as I understand should provide access to the server-side query)?

I ask this because browse layout (which requires size calculation) is delayed until the query is set, which is too late for me. This interferes with my "perform frame layout only once" changes, which requires that automatic frame layout is done only when the frame is defined. Even if the browse is defined without any columns (as in `def browse btt2 with size 10 by 10 title "btt2".`), setting its query later on does not affect its layout/columns.

So, if you don't have any objections, I'll move the logic in `Browse.setQuery` to `Browse.afterConfigUpdate` and leave only this:

```
public void setQuery(int queryId)
{
    if (browseModel != null)
        return;

    browseModel = new ClientBrowseModel(queryId);
}
```

The browse column will be retrieved via the widget ID, not query ID - because even if you change the query, the browse columns remain the same.

**#207 - 07/01/2015 09:00 AM - Stanislav Lomany**

Constantin, I have no objections.

**#208 - 07/01/2015 04:15 PM - Constantin Asofiei**

Stanislav, one more issue: is a little confusing that the query ID at some point is the same as the browse ID... this may vary at runtime (if the query is changed via the browse's QUERY attribute). We should make the query ID (BrowseConfig.queryId) reference the ID of the server-side QUERY resource.

**#209 - 07/01/2015 06:11 PM - Stanislav Lomany**

Yes, queryId actually contained browse ID. Didn't know that it can also contain query ID (I know that sounds strange).

We should make the query ID (BrowseConfig.queryId) reference the ID of the server-side QUERY resource.

Do we a reference to the query? I assumed that knowing browse ID is enough for the functionality. We have the reference to the query on the server side.

**#210 - 07/02/2015 02:38 AM - Stanislav Lomany**

Created task branch 2422b from P2J trunk revision 10893.

**#211 - 07/02/2015 03:50 AM - Constantin Asofiei**

Stanislav Lomany wrote:

Yes, queryId actually contained browse ID. Didn't know that it can also contain query ID (I know that sounds strange).

We should make the query ID (BrowseConfig.queryId) reference the ID of the server-side QUERY resource.

Do we a reference to the query? I assumed that knowing browse ID is enough for the functionality. We have the reference to the query on the server side.

What confuses me is the BrowseConfig.queryId field - this is set to the browse ID actually, on server-side... if all we need to reference the query on server-side is the browse ID, then we should remove this field and standardize any parameter names which, instead of a query ID, receive a browse ID (like ClientBrowseModel.queryId and others).



**#212 - 07/02/2015 07:45 AM - Stanislav Lomany**

I can do that after you finish your browse modifications if the issue doesn't bother you at this point.

**#213 - 07/02/2015 07:47 AM - Constantin Asofiei**

Stanislav Lomany wrote:

I can do that after you finish your browse modifications if the issue doesn't bother you at this point.

OK, that's good.

**#214 - 07/10/2015 05:44 PM - Stanislav Lomany**

At this point there are the following issues left:

1. In 4GL repositioned row is  $\leq$  DOWN value. DOWN value is calculated when the frame is created. The only exception I've found is when ROW-HEIGHT-CHARS attribute is specified (GUI). In this case DOWN value is recalculated when the frame is displayed. So Constantin's changes in frame layout timing may help.
2. REFRESHABLE issues. Repositioned row affect how browse behaves when REFRESHABLE is turned on.
3. MODIFIED attribute.

**#215 - 07/13/2015 11:51 AM - Stanislav Lomany**

Greg, about REFRESHABLE. Its typical use case is to turn it off - reposition - turn it on. However there is a potential case that a user can use browse while REFRESHABLE is off. Current P2J code tries to reproduce this behavior to a certain extent. But it is quirky, and with repositioned row it is even more quirky. So considering this complexity and that it is not likely to be used, I suggest to add it to the quirks list.

**#216 - 07/13/2015 12:05 PM - Greg Shah**

So considering this complexity and that it is not likely to be used, I suggest to add it to the quirks list.

OK.

**#217 - 07/15/2015 07:52 AM - Stanislav Lomany**

Greg, should I implement browse:MODIFIED attribute or column:MODIFIED too? I ask because they have completely different logic.

**#218 - 07/15/2015 08:17 AM - Greg Shah**

Please implement both.

## #219 - 07/16/2015 08:26 AM - Stanislav Lomany

Guys, about "transparent" data sending with AspectJ. P2J code calculates differences between "active (current) config" and "backup config" (not null) and sends them:

```
public WidgetConfigUpdates[] getConfigUpdates()
{
    ...
    WidgetConfig active = activeConfigs.get(wid);
    WidgetConfig backup = backupConfigs.get(wid);

    ...
    try
    {
        Object activeVal = wdef.fields[fid].get(active);
        Object backupVal = wdef.fields[fid].get(backup);

        boolean set = !((activeVal == null && backupVal == null) ||
            (activeVal != null && activeVal.equals(backupVal)));
    }
}
```

Consider I want to (re)register a widget with `ConfigManager.addWidgetConfig`. This config has boolean field. In the process field can be changed to true or false. So for any backup value (true/false) one of these cases fail - the "active" value is not sent because it matches "backup" value. Is there a work around for this?

You may say that at the point of widget registration its state should already be synchronized with server. But 1. this mechanism may contain bugs, I don't want to rely on it too much 2. I think I should have an ability to make force push at key points.

## #220 - 07/16/2015 09:09 AM - Hynek Cihlar

Stanislav Lomany wrote:

You may say that at the point of widget registration its state should already be synchronized with server.

Yes, this is the key. During registration the correct server value is persisted in the backup config and thus any deviation from the backup value will be properly synchronized to the server.

But 1. this mechanism may contain bugs, I don't want to rely on it too much 2. I think I should have an ability to make force push at key points.

If the mechanism contained bugs you would probably get ill behavior in between the forced config pushes anyway.

**#221 - 07/20/2015 05:24 AM - Stanislav Lomany**

Greg, for editing cells we have a single FillInWidget for each column. Its configuration is send to the client and used to create FillIns for in-cell editing. When we switch between cells that destroys one fill-in (its configuration is preserved) and creates a new fill-in re-using existing configuration. So we have two problems:

1. Widget re-creation resets its backup configuration. Backup configuration should match the previous state sent to the server, no the one that widget had on its recreation time. I can solve this by backing up and restoring backing configurations.
2. In some cases I have to update configuration for a non-current column, i.e. for a fill-in that doesn't exist on client side. This case be solved with an extra round trip to server.

OR I can solve both issues by creating a special set of configurations which will be unconditionally pushed to server as they needed. Which way do you think is better?

**#222 - 07/20/2015 09:19 AM - Stanislav Lomany**

Rebased task branch 2422b from P2J trunk revision 10896.

**#223 - 07/20/2015 09:54 AM - Greg Shah**

1. Widget re-creation resets its backup configuration. Backup configuration should match the previous state sent to the server, no the one that widget had on its recreation time. I can solve this by backing up and restoring backing configurations.
2. In some cases I have to update configuration for a non-current column, i.e. for a fill-in that doesn't exist on client side. This case be solved with an extra round trip to server.

I guess this sounds the least intrusive. Go with this approach.

**#224 - 07/20/2015 10:19 AM - Stanislav Lomany**

That is interesting:

```
** 007 IAS 20150604 Commented-out optimization in the getConfigUpdates() method as
**                  it looks to be not exactly correct
```

That means that the concept of "backup configs" is no longer in place. And I don't have to solve the problem 1. BUT if one day we'll decide that the ConfigManager code is correct, my code will definitely fail. On what assumption should I base my code?

**#225 - 07/20/2015 10:22 AM - Greg Shah**

It is OK to base your work on the unoptimized/current approach, but please:

1. Add a comment to the ConfigManager that highlights the browse code that will break if it is re-enabled.
2. Add a comment to the browse code that is dependent so that we can find it later. Make sure to explain the potential solution that will have to be implemented.

**#226 - 07/22/2015 11:44 AM - Stanislav Lomany**

Rebased task branch 2422b from P2J trunk revision 10898.

Please review task branch 2422b, revision 10906.

**#227 - 07/22/2015 12:30 PM - Greg Shah**

Code Review Task Branch 2422b Revision 10906

Eric: please review the query-related changes.

My only question: Does the 4GL browse-column only support MODIFIED for fill-in? I see that our support is limited in that way.

**#228 - 07/22/2015 12:35 PM - Stanislav Lomany**

As far as I tested, they match. Error message for browse column "Unable to assign UNKNOWN value to attribute MODIFIED on FILL-IN ..." suggests that it may be true.

**#229 - 07/22/2015 12:40 PM - Greg Shah**

Did you test with COMBO-BOX as a browse column? The error message may have only been formatted that way because the column was a fill-in.

**#230 - 07/22/2015 01:34 PM - Stanislav Lomany**

I'm not sure how to enable COMBO-BOX in browse. Even for GUI.

`DISPLAY column-list`

Specifies the column items to display in the browse. Note that the column-list cannot contain widgets other than fill-ins and the column-list cannot contain SKIP options.

**#231 - 07/22/2015 02:05 PM - Greg Shah**

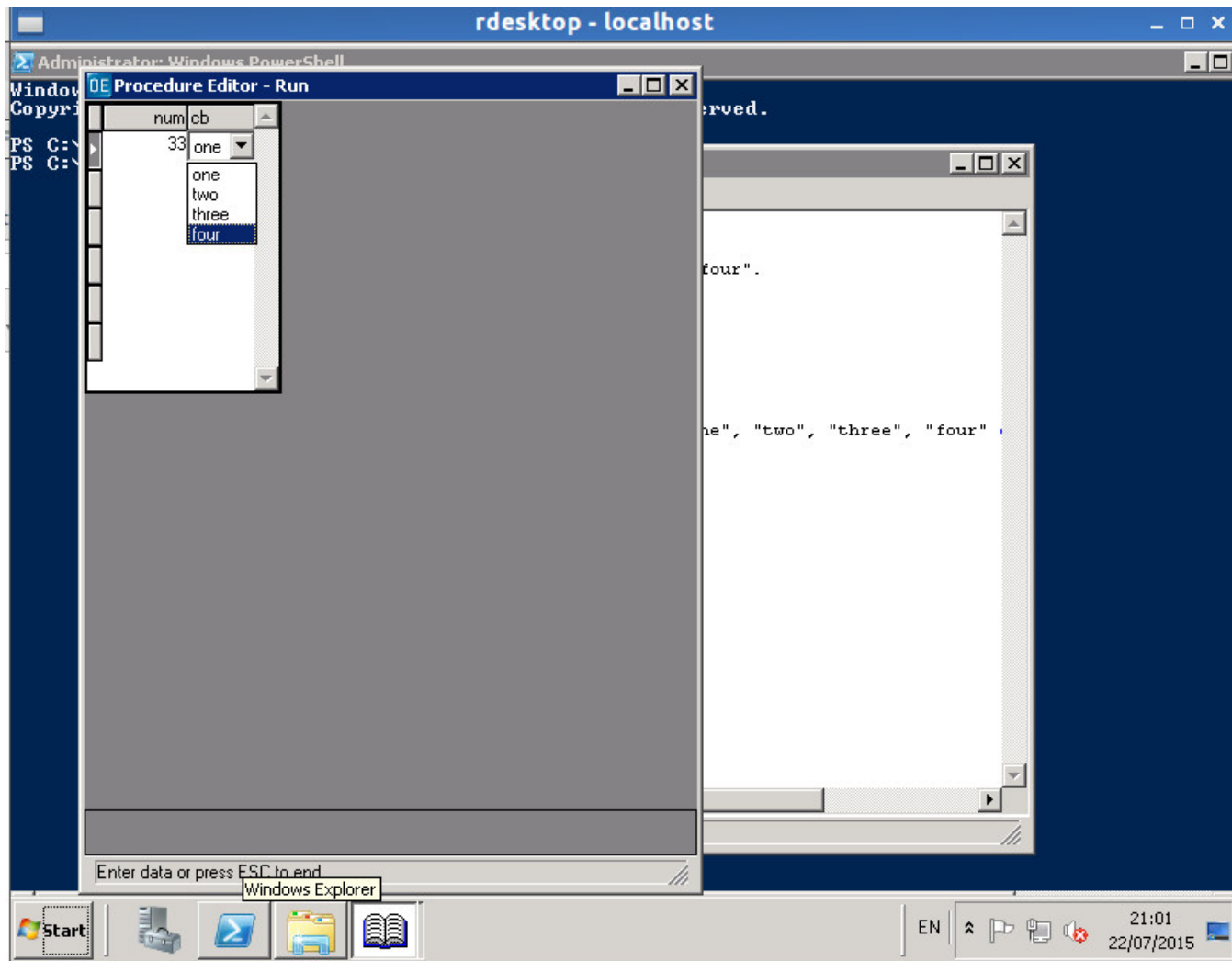
- *File combo\_box\_in\_browse\_does\_work.jpg added*

The 11.x documentation clearly states otherwise:

`DISPLAY column-list`

Specifies the column items to display in the browse. Note that the column-list cannot contain widgets other than fill-ins (default), combo-boxes, and toggle-boxes, and the column-list cannot contain SKIP options.

And it definitely does work:



Here is the sample code:

```
def temp-table tt
  field num as int
  field cb as char.

create tt.

def query q for tt.
open query q for each tt.

def browse b query q display tt.num
  tt.cb view-as combo-box inner-lines 5 list-items "one", "two", "three", "four"
  enable all with 10 down.

display b with frame f0.
enable b with frame f0.
wait-for go of frame f0.
```

The key is to use the VIEW-AS phrase inside the DEFINE BROWSE. If you put the same VIEW-AS phrase in the DEFINE TEMP-TABLE, it does not work.

**#232 - 07/22/2015 05:12 PM - Greg Shah**

You should start regression testing with the 10906 revision. We will continue discussing the browse-column support of other widget types, but we will check in the results if it passes and handle any further work later.

**#233 - 07/22/2015 05:34 PM - Eric Faulhaber**

Greg Shah wrote:

Code Review Task Branch 2422b Revision 10906

Eric: please review the query-related changes.

The logic looks OK to me, though changes to the behavior of Cursor always make me a bit nervous.

One minor note on format: please follow the coding standards w.r.t. enclosing single-line bodies of conditionals or loops in braces; i.e.:

```
if (foo)
{
    bar();
}
```

instead of:

```
if (foo)
    bar();
```

**#234 - 07/23/2015 08:03 AM - Greg Shah**

Code Review Task Branch 2422b Revision 10907

Everything looks good. If this passes testing, please check it in and notify the team.

I assume you are exploring the non-fill-in browse-column support. Please create a separate task to document the features that are missing and what changes are needed. Those will be worked later. Please make the new task related to this one.

**#235 - 07/26/2015 01:53 PM - Stanislav Lomany**

Please review task branch 2422b, revision 10908. Not yet merged with the latest baseline. I've changed approach of handling the case where the target reposition row doesn't match the actual one (e.g. if the target row was deleted).

**#236 - 07/27/2015 09:28 AM - Greg Shah**

Code Review Task Branch 2422b Revision 10908

I'm fine with the changes.

**#237 - 07/27/2015 05:06 PM - Stanislav Lomany**

Rebased task branch 2422b from P2J trunk revision 10901.

**#238 - 07/27/2015 05:53 PM - Greg Shah**

Code Review Task Branch 2422b Revision 10912

It looks fine. What is the status of regression testing?

**#239 - 07/27/2015 05:55 PM - Stanislav Lomany**

Just started it.

**#240 - 07/30/2015 04:33 AM - Stanislav Lomany**

*- Status changed from WIP to Review*

Branch 2422b has been committed to trunk revision 10902 (Implemented BROWSE:MODIFIED and SET-REPOSITIONED-ROW).

**#241 - 07/30/2015 07:54 AM - Greg Shah**

Please create the related task for non-fill-in column support in browse.

I know we have moved some GUI features to [#2564](#). And we moved quirks to [#2596](#).

Other than those above items, is there any work left for the items in this task? For example, the items in note 214? Or are there some TODOs left in the code somewhere? If there is any runtime functionality listed in this task that is not yet done, I need to know now.

**#242 - 07/30/2015 09:53 AM - Stanislav Lomany**

I cannot remember or see any unimplemented features. The only thing that comes to mind is that saving of new rows is not implemented. Currently we can insert them, but not save.

**#243 - 07/30/2015 09:58 AM - Greg Shah**

Stanislav Lomany wrote:

I cannot remember or see any unimplemented features. The only thing that comes to mind is that saving of new rows is not implemented. Currently we can insert them, but not save.

OK. Please create a related task for the save issue. It will be worked later. I'll close this task.

**#244 - 07/30/2015 10:46 AM - Greg Shah**

- Status changed from Review to Closed

**#245 - 11/16/2016 12:13 PM - Greg Shah**

- Target version changed from Milestone 12 to GUI Support for a Complex ADM2 App

**Files**

svl_upd20141102a.zip	427 KB	11/03/2014	Stanislav Lomany
svl_upd20141107a.zip	432 KB	11/07/2014	Stanislav Lomany
svl_upd20141107b.zip	432 KB	11/07/2014	Stanislav Lomany
svl_upd20141210a.zip	333 KB	12/10/2014	Stanislav Lomany
svl_upd20141216a.zip	69.4 KB	12/16/2014	Stanislav Lomany
svl_upd20141218a.zip	360 KB	12/19/2014	Stanislav Lomany
svl_upd20150225a.zip	366 KB	02/25/2015	Stanislav Lomany
svl_upd20150302a.zip	367 KB	03/02/2015	Stanislav Lomany
svl_upd20150305a.zip	279 KB	03/06/2015	Stanislav Lomany
svl_upd20150310a.zip	279 KB	03/10/2015	Stanislav Lomany
svl_upd20150311a.zip	281 KB	03/11/2015	Stanislav Lomany
svl_upd20150401a.zip	489 KB	03/31/2015	Stanislav Lomany
svl_upd20150403a.zip	490 KB	04/03/2015	Stanislav Lomany
svl_upd20150407a.zip	490 KB	04/08/2015	Stanislav Lomany
svl_upd20150415a.zip	277 KB	04/14/2015	Stanislav Lomany
svl_upd20150416a.zip	278 KB	04/17/2015	Stanislav Lomany
svl_upd20150503a.zip	351 KB	05/03/2015	Stanislav Lomany
svl_upd20150504a.zip	372 KB	05/04/2015	Stanislav Lomany
svl_upd20150505a.zip	373 KB	05/06/2015	Stanislav Lomany
svl_upd20150508a.zip	283 KB	05/08/2015	Stanislav Lomany
svl_upd20150608a.zip	358 KB	06/08/2015	Stanislav Lomany
svl_upd20150610a.zip	358 KB	06/10/2015	Stanislav Lomany
svl_upd20150612a.zip	358 KB	06/14/2015	Stanislav Lomany
svl_upd20150625a.zip	308 KB	06/25/2015	Stanislav Lomany
combo_box_in_browse_does_work.jpg	62.9 KB	07/22/2015	Greg Shah