

## Database - Bug #4450

### implement shared query and shared browse

12/06/2019 09:18 AM - Roger Borrello

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Stanislav Lomany	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>case_num:</b>	
<b>billable:</b>	No	<b>version:</b>	
<b>vendor_id:</b>	GCD		
<b>Description</b>			
<b>Related issues:</b>			
Related to User Interface - Bug #3592: shared frame interface needs to use th...			<b>New</b>

### History

#1 - 12/06/2019 09:40 AM - Roger Borrello

## Description

There are missing parameters in the generated registerQuery() as a result of a SHARED BROWSE

```
/home/rfb/testcases/uast/uast/buildarea/src/com/goldencode/testcases/abl/SharedQueryBrowseFrame.java:43: error
: illegal start of expression
    query0.registerQuery(, , 2, elementList0);
                        ^
/home/rfb/testcases/uast/uast/buildarea/src/com/goldencode/testcases/abl/SharedQueryBrowseFrame.java:43: error
: illegal start of expression
    query0.registerQuery(, , 2, elementList0);
                        ^
```

**When all instances of SHARED are removed, the code is generated correctly.**

There are 3 rules files that have reference to **registerQuery**:

```
/home/rfb/projects/fwd/4207a/rules/annotations/frame_scoping.rules
    this will emit a null second parameter for the registerQuery
/home/rfb/projects/fwd/4207a/rules/annotations/query_associations.rules
    this will emit a null first parameter for the registerQuery
/home/rfb/projects/fwd/4207a/rules/convert/ui_statements.rules
    <action>methodText = "registerQuery"</action>
    <!-- emit second parameter in registerQuery call -->
```

## Generated Code Snippets

### SharedQueryBrowserFrame.java

```
package com.goldencode.testcases.abl;

import com.goldencode.p2j.util.*;
import com.goldencode.testcases.ui.shared_frames.*;
import com.goldencode.p2j.ui.*;
import com.goldencode.p2j.persist.*;
import com.goldencode.testcases.dmo._temp.*;

import static com.goldencode.p2j.util.BlockManager.*;
```

```

/**
 * Business logic (converted to Java from the 4GL source code
 * in abl/shared_query_browse_frame.p).
 */
public class SharedQueryBrowseFrame
{
    Tt_1.Buf tt = (Tt_1.Buf) TemporaryBuffer.useShared("tt", Tt_1.Buf.class);

    Tt_1.Buf mybf = SharedVariableManager.lookupBuffer("mybf", "tt", Tt_1.Buf.class);

    Frame_1 myfrFrame = GenericFrame.importSharedFrame(Frame_1.class, "myfr");

    final QueryWrapper query0 = new QueryWrapper("myqr", true);

    /**
     * External procedure (converted to Java from the 4GL source code
     * in abl/shared_query_browse_frame.p).
     */
    public void execute()
    {
        character mytitle = UndoableFactory.character("Title");

        externalProcedure(SharedQueryBrowseFrame.this, new Block((Body) () ->
        {
            query0.addBuffer(mybf, true);
            RecordBuffer.openScope(tt, mybf);

            FrameElement[] elementList0 = new FrameElement[]
            {
                new Element(new FieldReference(mybf, "ttDesc"), myfrFrame.widgetMybrTtDescColumn()),
                new Element(new FieldReference(mybf, "ttAddr"), myfrFrame.widgetMybrTtAddrColumn())
            };
            query0.registerQuery(, , 2, elementList0);
        }));
    }
}

```

## Frame\_1.java

```

package com.goldencode.testcases.ui.shared_frames;

import com.goldencode.p2j.util.*;
import com.goldencode.p2j.ui.*;

public interface Frame_1
extends CommonFrame
{
    public character getMytitle();

    public void setMytitle(character parm);

    public void setMytitle(String parm);

    public void setMytitle(BaseDataType parm);

    public TextWidget widgetMytitle();
}

```

## Testcase

The below testcase is checked into testcase\_repo as uast/browse/shared\_query\_browse\_frame.p

```

define variable mytitle as character initial "Title".
define shared temp-table tt
    field tt-desc as character
    field tt-addr as character.

```

```
define shared buffer mybf for tt.
define shared query myqr for mybf scrolling.

define shared browse mybr query myqr
  display
    mybf.tt-desc format "x(30)" column-label "Description"
    mybf.tt-addr format "x(32)" label "Address"
  with 2 down.

define shared frame myfr
  mytitle no-label view-as text
  mybr
with
  centered no-label width 80 overlay
  title "Title".
```

Note that removing the mybf.tt-desc field from the browse results in a single error:

```
/home/rfb/testcases/uast/uast/buildarea/src/com/goldencode/testcases/abl/SharedQueryBrowseFrame.java:45: error
: illegal start of expression
    query0.registerQuery(, 2, elementList0);
                        ^
```

## Discussion (quote to continue)

**#3 - 12/06/2019 02:09 PM - Stanislav Lomany**

I check with you on the issue [#4450](#) to see if you have any help you can provide me to fix this.

Roger, it seems that the issue is about browse-related getters

```
public BrowseWidget widgetMybr();
```

```
public BrowseColumnWidget widgetMybrTtDescColumn();
public BrowseColumnWidget widgetMybrTtAddrColumn();
```

which are missing in the frame interface Frame\_1. After they're in place, the conversion issue may be resolved naturally.

#### #4 - 12/11/2019 11:31 AM - Roger Borrello

Stanislav Lomany wrote:

I check with you on the issue [#4450](#) to see if you have any help you can provide me to fix this.

Roger, it seems that the issue is about browse-related getters

[...]

which are missing in the frame interface Frame\_1. After they're in place, the conversion issue may be resolved naturally.

I'm having trouble with the testcase in 4GL. See #note-1 above for the testcase. 4GL syntax is correct, but running returns: **Unable to locate shared temp-table or work-table definition for table tt in procedure.**

#### #5 - 12/11/2019 11:33 AM - Constantin Asofiei

Roger, any external program which defines a simple shared resource needs to be invoked from another program which defines the resource as new shared. Having just shared and running the program directly will fail as the resource doesn't exist.

You need an external program which has define new shared for all define shared resource, and uses RUN statement to execute the other program.

#### #6 - 12/11/2019 11:40 AM - Roger Borrello

Constantin Asofiei wrote:

Roger, any external program which defines a simple shared resource needs to be invoked from another program which defines the resource as new shared. Having just shared and running the program directly will fail as the resource doesn't exist.

You need an external program which has define new shared for all define shared resource, and uses RUN statement to execute the other program.

Can that be done in an internal procedure?

**#7 - 12/11/2019 11:40 AM - Constantin Asofiei**

Roger Borrello wrote:

Constantin Asofiei wrote:

Roger, any external program which defines a simple shared resource needs to be invoked from another program which defines the resource as new shared. Having just shared and running the program directly will fail as the resource doesn't exist.

You need an external program which has define new shared for all define shared resource, and uses RUN statement to execute the other program.

Can that be done in an internal procedure?

No.

**#8 - 12/11/2019 12:05 PM - Roger Borrello**

Constantin Asofiei wrote:

Roger Borrello wrote:

Constantin Asofiei wrote:

Roger, any external program which defines a simple shared resource needs to be invoked from another program which defines the resource as new shared. Having just shared and running the program directly will fail as the resource doesn't exist.

You need an external program which has define new shared for all define shared resource, and uses RUN statement to execute the other program.

Can that be done in an internal procedure?

No.

Thanks... added uast/browse/shared\_query\_browse\_frame-run.p

**#9 - 12/11/2019 05:00 PM - Roger Borrello**

Stanislav Lomany wrote:

I check with you on the issue [#4450](#) to see if you have any help you can provide me to fix this.

Roger, it seems that the issue is about browse-related getters

[...]

which are missing in the frame interface `Frame_1`. After they're in place, the conversion issue may be resolved naturally.

This is the heart of the matter. I'm trying to find out why they aren't in `Frame_1`. My testing shows the scope of the `FRAME` being shared or not is the main issue. It doesn't matter if the `BROWSE` is shared or not, nor the `QUERY`.

I have been looking through annotations/frame\_scoping.rules for a clue.

**#10 - 12/12/2019 10:46 AM - Roger Borrello**

Just a little thing to mention, as I was looking through how orphan frames are created/detected:

There is a comment in `frame_generator_pre.xml`:

```
<!-- this is a hack to avoid frame interface computation more than once.
      TODO: 4GL allows the shared frame definition/structure to be split among as many
      UPDATE, FORM, DISPLAY, etc statements as it wants - we need to compute the frame
      structure from all of these, and not only from the first one -->
<rule>not fAlloc.isAnnotation("fr_interface")
  <variable name="formKey"           type="com.goldencode.p2j.uast.FrameAstKey" />
  <variable name="nextFrameSuffix"  type="java.lang.Integer" />
  <variable name="iName"            type="java.lang.String" />
  <variable name="biName"           type="java.lang.String" />
  <action>
    formKey = create("com.goldencode.p2j.uast.FrameAstKey", formNode, fName)
  </action>
  .
  .
  .
```

Does this pertain to this situation? Does the testcase build the frame definition/structure in a split manner?

Roger Borrello wrote:

Just a little thing to mention, as I was looking through how orphan frames are created/detected:

There is a comment in frame\_generator\_pre.xml:

[...]

Does this pertain to this situation? Does the testcase build the frame definition/structure in a split manner?

The creation of the Frame\_1.java is caused by annotations/frame\_generator\_pre.xml where the processForm function is receiving the parameter fName as null, and building the list of frameInterfaces with "Frame%s\_%d".

I am investigating how we end up there. The walk-rules are very short:

```
<!-- main processing (once per tree) -->
<walk-rules>
  <rule>type == prog.frame_alloc and
    # (boolean) copy.getAnnotation("shared")
    <action>unFormedFrames.add(copy)</action>
  </rule>
  <rule>type == prog.kw_form or
    (type == prog.define_frame and
      copy.getImmediateChild(prog.form_item, null) != null)
    <!-- basic name: extract frame name -->
    <action>println("frame_generator_pre (main 1): name=%s\n%s", this.text, this.dumpTree())</action>
    <action>frName = ""</action> <!-- default -->
    <action>ref = copy.getImmediateChild(prog.frame_phrase, null)</action>
    <rule>ref != null
      <action on="false">println("frame_generator_pre (main 2): ref.getImmediateChild(prog.frame_phrase, null) returns null")</action>
      <action>println("frame_generator_pre (main 2): name=%s\n%s", ref.text, ref.dumpTree())</action>
    </rule>
    <action>ref = ref.getImmediateChild(prog.kw_frame, null)</action>
    <rule>ref != null
      <action on="false">println("frame_generator_pre (main 3): ref.getImmediateChild(prog.kw_frame, null) returns null")</action>
      <action>println("frame_generator_pre (main 3): name=%s\n%s", ref.text, ref.dumpTree())</action>
    </rule>
    <action>ref = ref.getImmediateChild(prog.wid_frame, null)</action>
    <rule>ref != null
      <action on="false">println("frame_generator_pre (main 4): ref.getImmediateChild(prog.wid_frame, null) returns null")</action>
      <action>println("frame_generator_pre (main 4): name=%s\n%s", ref.text, ref.dumpTree())</action>
    </rule>
    <action>frName = ref.text</action>
  </rule>
</walk-rules>

<!-- detect if frame is SHARED (NEW or dependent), we skip independent frames for this analysis -->
<action>ref = getAst(#(java.lang.Long) copy.getAnnotation("frame-id"))</action>
<rule>#(boolean) ref.getAnnotation("shared")
  <action>execLib("processForm", copy, frName, ref)</action>
  <action>unFormedFrames.remove(ref)</action>
</rule>
</walk-rules>
```

The output from my execution reveals the attempt to get the child of the frame fails.

```
[java] -----
[java] Detecting Frame Interfaces
[java] -----
[java]
[java] ./abl/shared_query_browse_frame.p
```

```

[java] frame_generator_pre (main 1): name=define
[java] define [DEFINE_FRAME]:12884902003 @14:1
[java]   shared [KW_SHARED]:12884902005 @14:8
[java]   myfr [SYMBOL]:12884902009 @14:21
[java]   form item [FORM_ITEM]:12884902011 @0:0
[java]     expression [EXPRESSION]:12884902012 @0:0
[java]     mytitle [VAR_CHAR]:12884902013 @15:3
[java]     format phrase [FORMAT_PHRASE]:12884902015 @0:0
[java]       no-label [KW_NO_LABEL]:12884902016 @15:11
[java]       view-as [KW_VIEW_AS]:12884902018 @15:20
[java]       text [KW_TEXT]:12884902020 @15:28
[java]   form item [FORM_ITEM]:12884902022 @0:0
[java]     mybr [WID_BROWSE]:12884902023 @16:3
[java]     with [FRAME_PHRASE]:12884902025 @17:1
[java]       centered [KW_CENTER]:12884902027 @18:3
[java]       no-label [KW_NO_LABEL]:12884902029 @18:12
[java]       width [KW_WIDTH]:12884902031 @18:21
[java]       80 [NUM_LITERAL]:12884902033 @18:27
[java]       overlay [KW_OVERLAY]:12884902035 @18:30
[java]       title [KW_TITLE]:12884902037 @19:3
[java]       expression [EXPRESSION]:12884902039 @0:0
[java]       "Title" [STRING]:12884902040 @19:9
[java]
[java] frame_generator_pre (main 2): name=with
[java] with [FRAME_PHRASE]:12884902025 @17:1
[java]   centered [KW_CENTER]:12884902027 @18:3
[java]   no-label [KW_NO_LABEL]:12884902029 @18:12
[java]   width [KW_WIDTH]:12884902031 @18:21
[java]   80 [NUM_LITERAL]:12884902033 @18:27
[java]   overlay [KW_OVERLAY]:12884902035 @18:30
[java]   title [KW_TITLE]:12884902037 @19:3
[java]   expression [EXPRESSION]:12884902039 @0:0
[java]   "Title" [STRING]:12884902040 @19:9
[java]
[java] frame_generator_pre (main 3): ref.getImmediateChild(prog.kw_frame, null) returns null

```

I'm not sure that that logic is attempting to do, but it looks wrong.



**#12 - 12/12/2019 01:45 PM - Roger Borrello**

- Related to Bug #3592: shared frame interface needs to use the entire frame structure, not from each frame reference added

**#13 - 12/12/2019 02:30 PM - Roger Borrello**

Roger Borrello wrote:

I'm not sure that that logic is attempting to do, but it looks wrong.

The information in Constantin's updates to [#3592](#) are helpful, so I linked it to this issue for reference.

Could it be that this logic is *hard coded* to **FORM/FRAME\_PHRASE/KW\_FRAME/WID\_FRAME**? The **DEFINE\_FRAME** doesn't follow that pattern.

**#14 - 12/12/2019 03:29 PM - Roger Borrello**

Roger Borrello wrote:

Roger Borrello wrote:

I'm not sure that that logic is attempting to do, but it looks wrong.

The information in Constantin's updates to [#3592](#) are helpful, so I linked it to this issue for reference.

Could it be that this logic is *hard coded* to **FORM/FRAME\_PHRASE/KW\_FRAME/WID\_FRAME**? The **DEFINE\_FRAME** doesn't follow that pattern.

I made an update to `frame_generator_pre.xml` which now generates the class/Java name correctly, as `FrameMyfr_1`. However, the **BrowseWidget** and **BrowseColumnWidget** are still not being generated. The post-rules are still indicating there is an orphan SHARED FRAME.

**#15 - 12/13/2019 09:10 AM - Greg Shah**

Ovidiu/Stislav/Eric: Do we have any support for SHARED QUERY or SHARED BROWSE resources? I think the cause of this problem is simply that we have no support for either of those shared resources.

**#16 - 12/13/2019 09:24 AM - Stanislav Lomany**

I can't remember meeting a shared browse. And I didn't add any support for it.

**#17 - 12/13/2019 11:13 AM - Ovidiu Maxiniuc**

The queries also lack this features. The conversion ignores the new and shared options and no warning is issued whatsoever. I really was not aware of the fact that the queries (and browses) can be shared among procedures like the temp-tables.

**#18 - 12/16/2019 09:38 AM - Greg Shah**

- Assignee changed from Roger Borrello to Stanislav Lomany

- Subject changed from registerQuery missing parameters generated from SHARED BROWSE to implement shared query and shared browse

**#19 - 12/23/2019 09:54 AM - Stanislav Lomany**

Queries declared with DEF QUERY, e.g.

```
def new shared query q1-query for xtt1-table.
```

are converted to

```
final QueryWrapper query0 = new QueryWrapper("q1-query", false);
```

I'm going to convert it to

```
QueryWrapper query0 = SharedVariableManager.addQuery("q1-query", false);
```

for DEF NEW SHARED QUERY. And

```
QueryWrapper query0 = SharedVariableManager.lookupQuery("q1-query");
```

for DEF SHARED QUERY. I'm not sure if we need final modifcator for a shared query.

**#20 - 12/24/2019 11:16 AM - Stanislav Lomany**

Created task branch 4450a from FWD trunk revision 11339.

**#21 - 12/31/2019 10:00 AM - Roger Borrello**

Stanislav Lomany wrote:

Created task branch 4450a from FWD trunk revision 11339.

Let me know if you'd like me to run the customer code through this branch for testing purposes.

Question: Would you be able to make your modifications in 4207a (preferred) or are you planning to move them into trunk?

**#22 - 01/02/2020 05:09 AM - Stanislav Lomany**

Let me know if you'd like me to run the customer code through this branch for testing purposes.

Sure, as soon as the fix is ready.

Question: Would you be able to make your modifications in 4207a (preferred) or are you planning to move them into trunk?

I can merge them to 4207a after they're reviewed.

**#23 - 01/02/2020 06:59 AM - Greg Shah**

Please summarize the status of the task, what is left to do and the estimated time needed.

**#24 - 01/04/2020 03:35 PM - Stanislav Lomany**

4450a rev 11341 fixes the testcase, not sure about the customer's app. Please review.

**#25 - 01/06/2020 09:13 AM - Roger Borrello**

Stanislav Lomany wrote:

4450a rev 11341 fixes the testcase, not sure about the customer's app. Please review.

I will check, and return results. Thank you!

**#26 - 01/06/2020 09:19 AM - Roger Borrello**

Roger Borrello wrote:

Stanislav Lomany wrote:

4450a rev 11341 fixes the testcase, not sure about the customer's app. Please review.

I will check, and return results. Thank you!

Customer code still fails. I will revisit the testcase to see what is missing from it.

**#27 - 01/06/2020 11:18 AM - Greg Shah**

Code Review Task Branch 4450a Revision 11341

1. Why aren't shared browse instances registered with/imported from the SVM?
2. SVM needs javadoc for both addQuery() methods, the lookupQuery() method and the PersistentShares queries instance.
3. database\_access.rules, java\_templates.tpl and SVM are missing history entries.

**#28 - 01/07/2020 06:10 AM - Stanislav Lomany**

1. Why aren't shared browse instances registered with/imported from the SVM?

Browse widget is a part of a frame, so I assumed that browses are stored in SVM within a shared frame.

2. SVM needs javadoc for both addQuery() methods, the lookupQuery() method and the PersistentShares queries instance.
3. database\_access.rules, java\_templates.tpl and SVM are missing history entries.

Sorry, for some reason I assumed that the first commit contained javadocs and headers.

Issues in the customer's app are because of a regression caused by my changes. I found what changes caused it and I'm working on resolution.

**#29 - 01/07/2020 08:02 AM - Stanislav Lomany**

Heh, I forgot to define a local iterator and the one from the outer cycle was used. The files in the customer's app marked as related to [#4450](#) are now converted properly.

**#30 - 01/07/2020 08:11 AM - Greg Shah**

Please check in your changes and I will review. I'd like to have them merged into 4207a ASAP, if there is no issue.

1. Why aren't shared browse instances registered with/imported from the SVM?

Browse widget is a part of a frame, so I assumed that browses are stored in SVM within a shared frame.

I think this is probably an issue. The 4GL probably provides the ability to use a shared browse in more than one non-shared frame. Otherwise, why would a shared browse be needed?

Please check this and work it next. I want to get the current code into 4207a first. The next phase can be worked in 4207a directly, assuming that my theory is correct.

**#31 - 01/07/2020 09:20 AM - Stanislav Lomany**

- Status changed from New to WIP

Iterator issue and javadocs are fixed in 4450a rev 11342. (please note that I'll be unavailable for the next couple of hours)

**#32 - 01/07/2020 09:27 AM - Greg Shah**

Code Review Task Branch 4450a Revision 11342

I'm good with the changes. Please merge them into 4207a and then dead-archive 4450a.

**#33 - 01/07/2020 09:34 AM - Roger Borrello**

Greg Shah wrote:

Code Review Task Branch 4450a Revision 11342

I'm good with the changes. Please merge them into 4207a and then dead-archive 4450a.

I have also validated the previous version of customer code converts/compiles.

**#34 - 01/07/2020 06:05 PM - Stanislav Lomany**

4450a was merged into 4207a as rev 11371.

**#35 - 01/07/2020 06:57 PM - Roger Borrello**

Stanislav Lomany wrote:

4450a was merged into 4207a as rev 11371.

I updated my 4207a to that rev, and customer code compiled. Great job!

**#36 - 01/08/2020 05:13 AM - Stanislav Lomany**

The 4GL probably provides the ability to use a shared browse in more than one non-shared frame.

As far as I tested, I get compile-time or runtime error "BROWSE widget cannot be used in more than 1 frame".

Otherwise, why would a shared browse be needed?

I suppose it's just a marker. However I'm interested why, e.g. BUTTONs are not declared as shared.

**#37 - 01/08/2020 02:08 PM - Roger Borrello**

These testcases are broken:

```
./uast/browse/shared_query_browse_frame.p  
./uast/browse/shared_query_browse_frame-run.p
```

I am running using the latest 4207a-11372.

Stanislav, let's make sure your changes are in that branch properly.

**#38 - 01/08/2020 03:37 PM - Roger Borrello**

Roger Borrello wrote:

These testcases are broken:

```
./uast/browse/shared_query_browse_frame.p  
./uast/browse/shared_query_browse_frame-run.p
```

I am running using the latest 4207a-11372.

Stanislav, let's make sure your changes are in that branch properly.

I hadn't rebuild in my non-dev directory. I apologize for any churn.

#### #39 - 01/31/2020 09:31 AM - Greg Shah

Marian: Open questions which I'm hoping you can answer (see [#4450-36](#)):

- Is there any way to use a shared browse in more than one frame?
- If not, then what is the point? Is the shared keyword just ignored or does it have some real implementation?

#### #40 - 02/04/2020 12:08 AM - Roger Borrello

Stanislav, there is a change to rules/convert/frame\_generator.xml:

```
=== modified file 'rules/convert/frame_generator.xml'  
--- rules/convert/frame_generator.xml 2020-01-31 17:42:20 +0000  
+++ rules/convert/frame_generator.xml 2020-02-04 04:48:19 +0000  
@@ -4304,6 +4304,8 @@  
  
        save = #(com.goldencode.ast.Aast) brsMap.get(wname)  
    </action>  
  
+    <action>brsMap.remove(wname)</action>  
+  
  
    <!-- find DISPLAY phrase -->  
    <action>ref = save.getChildAt(0)</action>  
    <while>ref != null and ref.type != prog.kw_disp
```

The brsMap.remove(wname) line is removed from 4207a. I see a lot of duplicated widgets in many frame classes. When I add the line back in, the duplicate widgets are no longer generated. Is that removal of the widget from the brsMap something that is necessary?

**#41 - 02/04/2020 10:34 PM - Stanislav Lomany**

Roger, can you provide a testcase or point the files where duplicate widgets occur?  
I'm on vacation and most like I'll start fixing it around the end of the next week.

Is that removal of the widget from the brsMap something that is necessary?

It was necessary for shared browses conversion. But I'll find a way to fix the problem.

**#42 - 02/05/2020 07:33 AM - Marian Edu**

Greg Shah wrote:

Marian: Open questions which I'm hoping you can answer (see [#4450-36](#)):

- Is there any way to use a shared browse in more than one frame?

No, as soon as one try to add the same browse to another frame there is an error.

- If not, then what is the point? Is the shared keyword just ignored or does it have some real implementation?

Not really sure what is the point, haven't used that nor even think about doing it :)

However, the shared keyword is not ignored and the widget is indeed shared, apart the fact that you can't add it to another frame you can access it's properties and depending on it's state do everything it's possible to do with it - select a row, add/remove columns, it's just like you get a handle to the damn browse and work with it dynamically.

Mihai is working on some test cases, he will probably push those in bazaar today or tomorrow.

**#43 - 02/06/2020 09:27 PM - Roger Borrello**

Stanislav Lomany wrote:

Roger, can you provide a testcase or point the files where duplicate widgets occur?  
I'm on vacation and most like I'll start fixing it around the end of the next week.

Is that removal of the widget from the brsMap something that is necessary?



It was necessary for shared browses conversion. But I'll find a way to fix the problem.

There are a couple of good testcases. I sent you one directly, from customer code.

Another is `uast/browse/browse-at_base_field.p` which does not get frame-id annotations. One thing is the query and browse both have the same name, per the customer's code. A node is generated with annotations, but it is never grafted to the tree.

Finally, the original `uast/browse/shared_query_browse_frame-run.p` and `uast/browse/shared_query_browse_frame.p`

#### #44 - 02/18/2020 11:35 AM - Roger Borrello

Stanislav, do you have anything blocking you on getting 4207a into trunk? Please let us know what the status is. Thanks!

#### #45 - 02/18/2020 11:49 AM - Stanislav Lomany

Roger, I'm working on the regression caused by the shared browse changes. Regarding the at-field changes, I probably need the set of testcases which your changes are supposed to fix, specifically

```
+++ 199 RFB 20200106      Updated add_widget to be protective of situation where a widget
+++                          name referenced in an @ base-field doesn't result in a new widget.
+++ 050 RFB 20200106      There are situations where annotations from the parent instead of the
+++                          next non-aggregate is used, since there may not be any in that node.
+++                          Specifically when a widget name is referenced in an @ base-field.
```

#### #46 - 02/18/2020 12:30 PM - Roger Borrello

Stanislav Lomany wrote:

Roger, I'm working on the regression caused by the shared browse changes. Regarding the at-field changes, I probably need the set of testcases which your changes are supposed to fix, specifically

```
** 199 RFB 20200106      Updated add_widget to be protective of situation where a widget
**                          name referenced in an @ base-field doesn't result in a new widget.
```

This change was undone later. We implemented using another method of tracking when to add the @-base field as a widget to a frame.

```
** 050 RFB 20200106      There are situations where annotations from the parent instead of the
**                          next non-aggregate is used, since there may not be any in that node.
**                          Specifically when a widget name is referenced in an @ base-field.
```

This change in `screen_buffer.rules` was making sure the frame-id and the other widget annotations are placed in the correct node when there is an

@-base field. Previously it was only annotating the parent node.

The below are the testcases I used:

```
#!/uast/at/at_frame_mixup-shared.p
#!/uast/at/at_frame_mixup-shared-run.p

#!/uast/at/at_frame_broken_display-standalone.p
#!/uast/at/at_frame_broken_display2-standalone.p
#!/uast/at/at_frame_broken_display3-standalone.p
#!/uast/at/at_frame_broken_display4-standalone.p
#!/uast/at/at_frame_broken_display5-standalone.p
#!/uast/at/at_frame_broken_display6-standalone.p

#!/uast/at/at_base_field_max_quirky1.p

#!/uast/at/at_base_type_mismatch.p

#!/uast/at/at_faux_widget-simple.p
#!/uast/at/at_faux_widget-simple2.p

#!/uast/at/at_nonshared_frame_when.p
#!/uast/at/at_nonshared_frame_when2.p

#!/uast/at/at_shared_frames_broken.p
#!/uast/at/at_shared_frames_broken-run.p

#!/uast/at/at_frame_buffer-field.p
```

Have you updated your 4207a, since we did a rebase. If my changes are causing issues, please describe them, and I will help out.

**#47 - 02/18/2020 02:25 PM - Stanislav Lomany**

If my changes are causing issues, please describe them, and I will help out.

I was referencing the browse-at\_base\_field.p conversion problem.

#### #48 - 02/18/2020 02:48 PM - Roger Borrello

If you attempt to build ./uast/sharedframes/shared\_frame\_at\_label.p (below) and ./uast/sharedframes/shared\_frame\_at\_label-run.p on trunk, you will see a compilation error because the conversion does not create the myat widget on the shared frame. With my changes in 4207a, it is generated properly. Is that what you are looking for?

```
def shared frame sfl.  
def shared var fieldx as char no-undo.  
def shared var mymess as char init "Locale".
```

```
form fieldx  
  label "Label" at 2 myat as char no-label  
  with frame sfl.
```

```
display mymess @ myat with frame sfl.
```

```
message "Done.".
```

```
[javac] /home/rfb/projects/VirtualBox-VMs/shared/projects/testcases/src/com/goldencode/testcases/sharedframes/SharedFrameAtLabel.java:34: error: cannot find symbol  
[javac]         new Element(mymess, sflFrame.widgetMyat())  
[javac]                   ^  
[javac]    symbol:   method widgetMyat()  
[javac]    location: variable sflFrame of type FrameSfl_1  
[javac] 1 error
```

#### #49 - 02/18/2020 02:53 PM - Roger Borrello

Stanislav Lomany wrote:

If my changes are causing issues, please describe them, and I will help out.

I was referencing the browse-at\_base\_field.p conversion problem.

You can also compare results of trunk versus 4207a.

**#50 - 02/19/2020 11:54 AM - Roger Borrello**

Stanislav, saw your **4207a-rev 11398**, and thank you for that update! Are you able to help find the cause of the missing widget annotations as illustrated in `./uast/browse/browse-at_base_field.p`? It may not be directly related to shared browses but it would be very helpful if you could take a look.

Compare rev 11328 against current, and you will see we are keep track of any override frames (WITH FRAME frame) to be able to determine which frame a widget may be on, if there is not a frame specified. This is in `search_widget`. It may not be handling a browse.

Can you take a look? It's the last thing holding up a merge to trunk for 4207a.

**#51 - 02/19/2020 12:53 PM - Stanislav Lomany**

- *File difference.diff added*

Can you take a look?

Sure, but after I fix all of the `<chui_app>` regressions caused by shared browse changes.

I converted `<chui_app>` with 4207a-rev11398 and, except the shared browse issues, I found these differences. Not sure if all of them are expected or not (diff file is attached).

**#52 - 02/19/2020 12:55 PM - Greg Shah**

- *File deleted (difference.diff)*

**#53 - 02/19/2020 01:34 PM - Roger Borrello**

I looked and the testcase using the debug version of `frame_scoping.rules` shows the override frame on line 24 of the testcase is not being used. For some reason, the global variable `override_frame_blockid` does not maintain the frame-id once found.

**#54 - 02/24/2020 01:51 AM - Stanislav Lomany**

4207a rev 11402 has the shared browse issues fixed: conversion regression testing passes. BTW runtime regression testing seem to be failing for this branch.

I'm going to look into 'frame-id' annotation problem.

**#55 - 02/24/2020 07:28 AM - Roger Borrello**

Stanislav Lomany wrote:

4207a rev 11402 has the shared browse issues fixed: conversion regression testing passes. BTW runtime regression testing seem to be failing

for this branch.  
I'm going to look into 'frame-id' annotation problem.

Is that the same as the issue converting `./uast/browse/browse-at_base_field`?

**#56 - 02/24/2020 02:06 PM - Stanislav Lomany**

Is that the same as the issue converting `./uast/browse/browse-at_base_field`?

Yes, my main goal is to fix this testcase. Also conversion of a customer app fails because of the same issue. I suppose that'll help merging 4207a into trunk.

**#57 - 02/26/2020 08:42 AM - Stanislav Lomany**

This change in `screen_buffer.rules` was making sure the `frame-id` and the other widget annotations are placed in the correct node when there is an `@-base` field. Previously it was only annotating the parent node.

The below are the testcases I used:

- `./uast/at/at_frame_mixup-shared.p`
- `./uast/at/at_frame_mixup-shared-run.p`
- `./uast/at/at_frame_broken_display-standalone.p`
- `./uast/at/at_frame_broken_display2-standalone.p`
- `./uast/at/at_frame_broken_display3-standalone.p`
- `./uast/at/at_frame_broken_display4-standalone.p`
- `./uast/at/at_frame_broken_display5-standalone.p`
- `./uast/at/at_frame_broken_display6-standalone.p`
- `./uast/at/at_base_field_max_quirky1.p`
- `./uast/at/at_base_type_mismatch.p`
- `./uast/at/at_faux_widget-simple.p`
- `./uast/at/at_faux_widget-simple2.p`
- `./uast/at/at_nonshared_frame_when.p`
- `./uast/at/at_nonshared_frame_when2.p`
- `./uast/at/at_shared_frames_broken.p`
- `./uast/at/at_shared_frames_broken-run.p`
- `./uast/at/at_frame_buffer-field.p`

Roger, I see that the 'frame-id' problem appeared after you started to get this annotation from "format" node instead of "expression" node in the "EXPRESSION AT FORMAT" statement:

```
<action on="false">annref = source.parent</action>
..
<action>annref = fmtref</action> <-----
```

Is "format" node supposed to have this annotation? Did you do any changes that assigned this annotation to this node? Could you advise what testcase needs the `annref = fmtref` change? Can I find this testcase among the list of testcase you provided? BTW most of

these testcases are not checked into bzd.

#### #58 - 02/26/2020 09:32 AM - Roger Borrello

Stanislav Lomany wrote:

This change in screen\_buffer.rules was making sure the frame-id and the other widget annotations are placed in the correct node when there is an @-base field. Previously it was only annotating the parent node.

The below are the testcases I used:

```
#!/uast/at/at_frame_mixup-shared.p
#!/uast/at/at_frame_mixup-shared-run.p
#!/uast/at/at_frame_broken_display-standalone.p
#!/uast/at/at_frame_broken_display2-standalone.p
#!/uast/at/at_frame_broken_display3-standalone.p
#!/uast/at/at_frame_broken_display4-standalone.p
#!/uast/at/at_frame_broken_display5-standalone.p
#!/uast/at/at_frame_broken_display6-standalone.p
#!/uast/at/at_base_field_max_quirky1.p
#!/uast/at/at_base_type_mismatch.p
#!/uast/at/at_faux_widget-simple.p
#!/uast/at/at_faux_widget-simple2.p
#!/uast/at/at_nonshared_frame_when.p
#!/uast/at/at_nonshared_frame_when2.p
#!/uast/at/at_shared_frames_broken.p
#!/uast/at/at_shared_frames_broken-run.p
#!/uast/at/at_frame_buffer-field.p
```

Roger, I see that the 'frame-id' problem appeared after you started to get this annotation from "format" node instead of "expression" node in the "EXPRESSION AT FORMAT" statement:

[...]

Is "format" node supposed to have this annotation? Did you do any changes that assigned this annotation to this node?

Could you advise what testcase needs the annref = fmtref change? Can I find this testcase among the list of testcase you provided? BTW most of these testcases are not checked into bzd.

That change was left behind mistakenly. It was supposed to be backed out when I "undid" the @-base field changes I had made. I missed that. I am so thankful for your discovery! Would you like me to handle the change to screen\_buffer.rules, or can you make the changes?

I also checked in those missing testcases.

**#59 - 02/26/2020 09:34 AM - Stanislav Lomany**

Would you like me to handle the change to screen\_buffer.rules,

That seems a safer option to me.

**#60 - 02/26/2020 09:43 AM - Roger Borrello**

Stanislav Lomany wrote:

Would you like me to handle the change to screen\_buffer.rules,

That seems a safer option to me.

Checked into **4207a-11410**. Can you do regression testing?

**#61 - 02/26/2020 10:21 AM - Stanislav Lomany**

Can you do regression testing?

Sure.

**#62 - 03/03/2020 01:02 PM - Greg Shah**

- % Done changed from 0 to 100

- Status changed from WIP to Closed

**#63 - 03/03/2020 01:04 PM - Greg Shah**

Branch 4207a was merged to trunk as revision 11344.