

User Interface - Bug #2778

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

./ask-gui.p fails if "Program name" is filled with a misspelled program name

10/23/2015 09:47 AM - Sergey Ivanovskiy

Status:	Closed	Start date:	10/23/2015
Priority:	Normal	Due date:	
Assignee:	Sergey Ivanovskiy	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	GUI Support for a Complex ADM2 App	case_num:	
billable:	No		
vendor_id:	GCD		
Description			
Related issues:			
Related to Runtime Infrastructure - Bug #2860: implement proper server certif...			Closed
Related to User Interface - Bug #2886: missing pause when an invalid program ...			New

History

#1 - 10/23/2015 09:49 AM - Sergey Ivanovskiy

If "Program name" is filled with a misspelled program name, then the swing/web client fails and the swing client restarts unresponsive. The swing client log contains the following exception:

```
Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
    at com.goldencode.p2j.ui.client.gui.driver.swing.SwingEmulatedWindow$3.windowLostFocus (SwingEmulatedWindow.java:269)
    at java.awt.Window.processWindowFocusEvent (Window.java:2108)
    at java.awt.Window.processEvent (Window.java:2021)
    at java.awt.Component.dispatchEventImpl (Component.java:4881)
    at java.awt.Container.dispatchEventImpl (Container.java:2292)
    at java.awt.Window.dispatchEventImpl (Window.java:2750)
    at java.awt.Component.dispatchEvent (Component.java:4703)
    at java.awt.KeyboardFocusManager.redispatchEvent (KeyboardFocusManager.java:1954)
    at java.awt.DefaultKeyboardFocusManager.typeAheadAssertions (DefaultKeyboardFocusManager.java:995)
    at java.awt.DefaultKeyboardFocusManager.dispatchEvent (DefaultKeyboardFocusManager.java:749)
    at java.awt.Component.dispatchEventImpl (Component.java:4752)
    at java.awt.Container.dispatchEventImpl (Container.java:2292)
    at java.awt.Window.dispatchEventImpl (Window.java:2750)
    at java.awt.Component.dispatchEvent (Component.java:4703)
    at java.awt.EventQueue.dispatchEventImpl (EventQueue.java:758)
    at java.awt.EventQueue.access$500 (EventQueue.java:97)
    at java.awt.EventQueue$3.run (EventQueue.java:709)
    at java.awt.EventQueue$3.run (EventQueue.java:703)
    at java.security.AccessController.doPrivileged (Native Method)
    at java.security.ProtectionDomain$1.doIntersectionPrivilege (ProtectionDomain.java:75)
    at java.security.ProtectionDomain$1.doIntersectionPrivilege (ProtectionDomain.java:86)
    at java.awt.EventQueue$4.run (EventQueue.java:731)
    at java.awt.EventQueue$4.run (EventQueue.java:729)
    at java.security.AccessController.doPrivileged (Native Method)
    at java.security.ProtectionDomain$1.doIntersectionPrivilege (ProtectionDomain.java:75)
    at java.awt.EventQueue.dispatchEvent (EventQueue.java:728)
    at java.awt.SequencedEvent.dispatch (SequencedEvent.java:128)
    at java.awt.EventQueue.dispatchEventImpl (EventQueue.java:756)
    at java.awt.EventQueue.access$500 (EventQueue.java:97)
    at java.awt.EventQueue$3.run (EventQueue.java:709)
    at java.awt.EventQueue$3.run (EventQueue.java:703)
    at java.security.AccessController.doPrivileged (Native Method)
    at java.security.ProtectionDomain$1.doIntersectionPrivilege (ProtectionDomain.java:75)
    at java.security.ProtectionDomain$1.doIntersectionPrivilege (ProtectionDomain.java:86)
    at java.awt.EventQueue$4.run (EventQueue.java:731)
    at java.awt.EventQueue$4.run (EventQueue.java:729)
    at java.security.AccessController.doPrivileged (Native Method)
```

```

at java.security.ProtectionDomain$1.doIntersectionPrivilege(ProtectionDomain.java:75)
at java.awt.EventQueue.dispatchEvent(EventQueue.java:728)
at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:201)
at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:116)
at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:105)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:101)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:93)
at java.awt.EventDispatchThread.run(EventDispatchThread.java:82)

```

#2 - 10/23/2015 02:55 PM - Greg Shah

- Assignee set to *Sergey Ivanovskiy*
- Target version set to *Milestone 12*

#3 - 10/26/2015 09:15 AM - Sergey Ivanovskiy

- File *ask-gui.png* added

If the input program is misspelled, then Progress 4GL displays the "was not found" error message and completes ask-gui.p after the error message is acknowledged.

#4 - 10/27/2015 06:30 AM - Sergey Ivanovskiy

There are two issues here. The first one is the corresponding error dialog isn't shown and the second one is pauseBeforeEnd isn't triggered. The dialog must be displayed on public void displayError(String errmsg) of ErrorWriterInteractive. It follows that the current implementation of this method doesn't show any dialogs. But there is public void messageBox(String masterMsg, String title) that displays the target dialog. The second issue is unclear for me, because StopConditionException doesn't break the "do while again" loop and finally is caught by StandardServer public static Object invoke(int stopDisp, Isolatable entry) with the following action LogicalTerminal.setRelogin(true);

```

Object result = null;
try
{
    startup:
    {
        TransactionManager.blockSetup();

        do
        {
            try
            {
                // calling the entry point
                result = entry.execute();
            }

            catch (StopConditionException sce)
            {
                if (stopDisp != 0)
                {
                    if (stopDisp == 1)
                        LogicalTerminal.setRelogin(true);

                    throw sce;
                }

                TransactionManager.disableLoopProtection(null);
                TransactionManager.triggerRetry(null);
            }

            catch (ConditionException ce)
            {
                if (ce instanceof QuitConditionException)

```

```
    {
        TransactionManager.setProcessingQuit(true);
    }
    break startup;
}
```

.....

and is followed by the application restart.

#5 - 10/27/2015 09:59 AM - Sergey Ivanovskiy

- File *displayError_1.txt* added

If `stop_disposition` is set to 2 then the restart doesn't happen.

```
<node class="integer" name="stop_disposition">
  <node-attribute name="value" value="2"/>
</node>
```

Greg, please review this diff. I have checked native chui, swing chui, web chui, web gui and swing gui drivers types and it seems to work with `stop_disposition = 2` according to Progress 4GL.

#6 - 10/27/2015 11:13 AM - Sergey Ivanovskiy

- File *invalid_chars.png* added

But these changes lead to the errors are displayed as message boxes. For example, running this program `./window_sizing/default_empty_window.p` shows numerous messages of "Invalid character unit value ..." that have been displayed as messages in the status area before these changes.

#7 - 10/27/2015 11:33 AM - Greg Shah

But these changes lead to the errors are displayed as message boxes. For example, running this program `./window_sizing/default_empty_window.p` shows numerous messages of "Invalid character unit value ..." that have been displayed as messages in the status area before these changes.

Yes, the change is incorrect.

Consider this (save it to a file named `gen_err.p`):

```
DEF VAR i AS INT.

MESSAGE "1".
i = INTEGER("n2").
MESSAGE "2".
```

If you run in on windev01, using `prowin32 -p gen_err.p`, then you get this:

`gen_err.p_does_not_use_alert_box_in_progress_gui_20151027.png`

I think you have been confused by running your testcase using the procedure editor. If you do that, it will display in an alert-box.

Also, I don't understand how the error message display is related to the NPE in note 1. Please help me understand the connection.

The second issue is unclear for me, because `StopConditionException` doesn't break the "do while again" loop and finally is caught by `StandardServer`

`public static Object invoke(int stopDisp, Isolatable entry)` with the following action `LogicalTerminal.setRelogin(true)`;

The default behavior of restarting on a stop condition in P2J is actually how the 4GL works, even in GUI. The implementation you found is intentional. Also, as you found, it is configurable for each installation.

Consider this code (save it as `gen_stop.p`):

```
DEF VAR dt AS DATETIME INIT NOW.  
MESSAGE "Attempt at " + string(dt).  
RUN does_not_exist.p.  
MESSAGE "After " + STRING(dt).
```

Make sure you run this with `prowin32 -p gen_stop.p`. DO NOT run this from the procedure editor. The procedure editor changes the execution state for error processing and many other factors that can be misleading.

This code will re-start continually (note that the 2nd message is never displayed and the timestamp constantly increases after each space bar press). You can use ESC to exit.

I don't think there is an issue with the stop disposition in P2J.

#8 - 10/27/2015 01:52 PM - Sergey Ivanovskiy

Yes, you are right, I am running programs from the procedure editor and it confuses me. NPE is related to `ThinClient.getInstance()` returns null. Thus it needs to investigate why this call returns null object and why the client becomes unresponsive.

#9 - 10/28/2015 04:57 AM - Sergey Ivanovskiy

After the swing client has been restarted, it has two instances of `WinKeyboardReader`, the old one handles keys input and the new one is used by the recreated `KeyReader` thread. Planning to fix it today.

For `ChuiSwingDriver` this bug isn't reproduced due to it doesn't use `ContextLocal` storage for `SwingKeyboardReader`

```
public int readKey()
{
    return pane.readKey();
}
```

In the case of `SwingGuiDriver` the keyboard reader is saved in the storage context `keyReader`

```
public int readKey()
{
    return keyReader.get().read();
}
```

and since the local storage context is cleared before the new client session has been established, the new instance of `WinKeyboardReader` is created. But the driver instance is the same and it isn't recreated. NPE is due to the driver isn't recreated and its focus listener that is invoked by the separate swing UI thread requires the instance of `ThinClient` in the middle of its creation.

#10 - 10/28/2015 06:11 AM - Sergey Ivanovskiy

- *File restart_1.txt added*

Greg, we can recreate the driver instance if the application requires to be restarted. I have checked it works. Do you agree?

#11 - 10/28/2015 07:37 AM - Sergey Ivanovskiy

Is it correct that `WindowManager` on public static void `clearWindowList()` doesn't invoke the driver method public void `deregisterWindow(int windowId)` for the cleared windows? Also widget state listeners of `WindowGuiImpl` are not invoked

```
registerWidgetStateListener(new WidgetStateListener()
{
    @Override
    public void notifyWidgetDestroyed(Widget<? extends OutputManager<?>> w)
    {
        FontManager.deregisterFontTable(WindowGuiImpl.this, OutputManager.getDriver());
        gd.deregisterWindow(w.getId().asInt());
    }
});
```

```
}
```

#12 - 10/28/2015 08:02 AM - Greg Shah

we can recreate the driver instance if the application requires to be restarted. I have checked it works. Do you agree?

In the original design we intentionally reused the driver instance. I worry that we may have dependencies that are not obvious.

For example, in the web client could this cause the jetty port to change in the case where we are using a random port number (instead of one that is hard coded in the directory).

Just setting the driver = null on every pass through the loop doesn't seem like it would help anyway, since the driver instance is cached in the OutputManager.

#13 - 10/28/2015 08:04 AM - Greg Shah

Constantin/Hynek: perhaps you have thoughts on the question in note 11 about the WindowManager.

#14 - 10/28/2015 09:45 AM - Sergey Ivanovskiy

we can recreate the driver instance if the application requires to be restarted. I have checked it works. Do you agree?

In the original design we intentionally reused the driver instance. I worry that we may have dependencies that are not obvious.

For example, in the web client could this cause the jetty port to change in the case where we are using a random port number (instead of one that is hard coded in the directory).

Just setting the driver = null on every pass through the loop doesn't seem like it would help anyway, since the driver instance is cached in the OutputManager.

Another way to remove listeners is to invoke `deregisterWindow(int windowId)`. But it appears new issue the window we need to remove has `WidgetId.DEFAULT_WINDOW_ID`. Please look at `WindowManager`

```
/**
 * Remove the window instance from the client.
 *
 * @param window
 *       The window ID.
 */
public static void removeWindow(final int window)
{
    TitledWindow<?> w = (TitledWindow<?>) WindowManager.findWindow(window);

    // the default window will be deleted only when the client exits/gets reset
    if (w != null && !w.config().id.equals(WidgetId.DEFAULT_WINDOW_ID))
```

```
{  
    // dynamic window, remove it from the window manager  
    WindowManager.remove(w);  
}  
}
```

#15 - 10/28/2015 10:05 AM - Sergey Ivanovskiy

- File *restart_2.txt* added

Please review this attached fix.

#16 - 10/28/2015 02:28 PM - Greg Shah

My only concern is that we would probably also have to call `WindowManager.reset()` so that the context-local `defaultWnd` and `currentWnd` will be cleared too. Otherwise it seems like we would retain references to the old default-window. In GUI, we already seem to retain references to the old current-window, which seems wrong.

We may have some dependencies on this that would have to be cleaned up. In other words: I wonder why we don't already call `WindowManager.reset()` in GUI. It may cause a problem.

Constantin/Hynek: do either of you have concerns with allowing the `WindowManager.clearWindowList()` to remove the default-window?

#17 - 10/28/2015 02:37 PM - Hynek Cihlar

Greg Shah wrote:

Constantin/Hynek: perhaps you have thoughts on the question in note 11 about the `WindowManager`.

I would say it is not correct that `deregisterWindow()` is not called for the cleared windows. But it probably doesn't matter much as `WindowManager.clearWindowList()` is only called from `ThinClient.terminate()`. Sergey, what is your use case?

#18 - 10/28/2015 02:43 PM - Greg Shah

Just run `ask-gui.p` and enter a non-existing-file-name.p.

#19 - 10/28/2015 02:56 PM - Sergey Ivanovskiy

Yes, `ask-gui.p` with any non existing program name.

#20 - 10/28/2015 03:01 PM - Sergey Ivanovskiy

This test case also raises another bug. If the program restarts, the input is sometimes blocked on public synchronized `KeyInput.getKeystroke(int msec, boolean honorServerEvents)` of `TypeAhead` and it is unblocked after the main window is collapsed and then activated again. It raises OS events and may be unblocks it.

#21 - 10/28/2015 03:12 PM - Hynek Cihlar

Greg Shah wrote:

My only concern is that we would probably also have to call `WindowManager.reset()` so that the context-local `defaultWnd` and `currentWnd` will be cleared too. Otherwise it seems like we would retain references to the old default-window. In GUI, we already seem to retain references to the old current-window, which seems wrong.

We may have some dependencies on this that would have to be cleaned up. In other words: I wonder why we don't already call `WindowManager.reset()` in GUI. It may cause a problem.

Constantin/Hynek: do either of you have concerns with allowing the `WindowManager.clearWindowList()` to remove the default-window?

My concern is that there are some differences in how the default window is being initialized. If you plan to reset the default window by destroying and recreating it, you may hit on some hard assumptions in the init logic.

#22 - 10/28/2015 03:20 PM - Sergey Ivanovskiy

My concern is that there are some differences in how the default window is being initialized. If you plan to reset the default window by destroying and recreating it, you may hit on some hard assumptions in the init logic.

Please could you clarify the differences in how the default window is being initialized.

#23 - 10/28/2015 03:29 PM - Sergey Ivanovskiy

If we save the default window after the client has been restarted, we need to save the storage context, but the last one is clean (please see ClientCore line 298 the current branch 2677a) by `SecurityManager.clearInstance()`.

#24 - 10/28/2015 03:56 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

My concern is that there are some differences in how the default window is being initialized. If you plan to reset the default window by destroying and recreating it, you may hit on some hard assumptions in the init logic.

Please could you clarify the differences in how the default window is being initialized.

I think you will get the most accurate answer if you read the code. You can start by placing a breakpoint in `WindowGuiImpl.ctor()`.

#25 - 10/28/2015 04:39 PM - Greg Shah

My concern is that there are some differences in how the default window is being initialized. If you plan to reset the default window by destroying and recreating it, you may hit on some hard assumptions in the init logic.

Hynek: do you know of some other way the default-window gets initialized except through `WindowManager.getDefaultWindow()`?

Is there any difference between ChUI and GUI in regards to default-window creation?

#26 - 10/28/2015 04:56 PM - Hynek Cihlar

Greg Shah wrote:

My concern is that there are some differences in how the default window is being initialized. If you plan to reset the default window by destroying and recreating it, you may hit on some hard assumptions in the init logic.

Hynek: do you know of some other way the default-window gets initialized except through `WindowManager.getDefaultWindow()`?

No.

Is there any difference between ChUI and GUI in regards to default-window creation?

I don't recall any differences except the obvious - different UI-specific classes.

#27 - 10/28/2015 05:41 PM - Greg Shah

Sergey Ivanovskiy wrote:

This test case also raises another bug. If the program restarts, the input is sometimes blocked on public synchronized KeyInput getKeyStroke(int msec, boolean honorServerEvents) of TypeAhead and it is unblocked after the main window is collapsed and then activated again. It raises OS events and may be unblocks it.

I assume it is blocked in wait()?

The thread calling TypeAhead.getKeyStroke() will be the main client thread. It will call there if and only if the client is waiting for user input in ThinClient.waitForEvent() or ThinClient.messageBoxEventWorker(). While there are events to be processed, the thread will not wait() in getKeyStroke().

If there are no events, then the call from ThinClient.waitForEvent() could wait an indefinite amount of time if it was called with seconds == 0.

The only obvious problem is if there was a way to call TypeAhead.closeReader() before the wait() occurred OR while the KeyReader.run() was blocked in mgr.readKey(). In the first case, there would either be a call to notify() while there was no waiter, so the notify would be lost and any subsequent wait() would never be unblocked. In the second case, the KeyReader.run() would exit without calling notify() leaving any waiter unnotified forever.

However, neither of those cases should be possible, since the thread calling closeReader() should be the main thread and it would be in the middle of terminate(), so it should not then call back into getKeyStroke().

Do you have a stack trace from this deadlock condition?

#28 - 10/28/2015 05:43 PM - Greg Shah

Hynek Cihlar wrote:

Greg Shah wrote:

My concern is that there are some differences in how the default window is being initialized. If you plan to reset the default window by destroying and recreating it, you may hit on some hard assumptions in the init logic.

Hynek: do you know of some other way the default-window gets initialized except through WindowManager.getDefaultWindow()?

No.

Is there any difference between ChUI and GUI in regards to default-window creation?

I don't recall any differences except the obvious - different UI-specific classes.

Sergey: make your change to clear the default-window. You should test it with all the different clients, both ChUI and GUI.

#29 - 10/28/2015 05:44 PM - Greg Shah

Sergey: make your change to clear the default-window. You should test it with all the different clients, both ChUI and GUI.

Don't commit your changes to 2677a. Just do your testing and report results back here.

#30 - 10/29/2015 03:43 AM - Sergey Ivanovskiy

Greg Shah wrote:

Sergey Ivanovskiy wrote:

This test case also raises another bug. If the program restarts, the input is sometimes blocked on public synchronized KeyInput getKeystroke(int msec, boolean honorServerEvents) of TypeAhead and it is unblocked after the main window is collapsed and then activated again. It raises OS events and may be unblocks it.

I assume it is blocked in wait()?

The thread calling TypeAhead.getKeystroke() will be the main client thread. It will call there if and only if the client is waiting for user input in ThinClient.waitForEvent() or ThinClient.messageBoxEventWorker(). While there are events to be processed, the thread will not wait() in getKeystroke().

If there are no events, then the call from ThinClient.waitForEvent() could wait an indefinite amount of time if it was called with seconds == 0.

The only obvious problem is if there was a way to call TypeAhead.closeReader() before the wait() occurred OR while the KeyReader.run() was blocked in mgr.readKey(). In the first case, there would either be a call to notify() while there was no waiter, so the notify would be lost and any subsequent wait() would never be unblocked. In the second case, the KeyReader.run() would exit without calling notify() leaving any waiter unnotified forever.

However, neither of those cases should be possible, since the thread calling closeReader() should be the main thread and it would be in the middle of terminate(), so it should not then call back into getKeystroke().

Do you have a stack trace from this deadlock condition?

Greg, excuse me for this issue was expressed unclear. It is not a deadlock of two threads that are waiting for one another. It is the issue when the focused window doesn't get keys input until it is collapsed and activated again. I am trying to resolve the issue but the breakpoints method doesn't help here because the issue disappears if the client window is activated repeatedly.

#31 - 10/29/2015 08:59 AM - Sergey Ivanovskiy

The issue with blocking keys input is due to the fact that the current keyboard focus manager has not have an active window. In order to prove it I have inserted this log code on mouse click event

```
KeyboardFocusManager fm = KeyboardFocusManager.getCurrentKeyboardFocusManager();
System.err.println("active window " + fm.getActiveWindow());
System.err.println("focused window " + fm.getFocusedWindow());
System.err.println("focus owner " + fm.getFocusOwner());
```

and finally it is reported null references in the case of the unresponsive keys input. After the swing client window has been minimized and restored, these references are set to the SwingGuiClient instance.

#32 - 10/30/2015 01:53 PM - Sergey Ivanovskiy

- File restart_4.txt added

Greg, please review this work around. I am planning to work on the better solution, there is an idea to use KeyEventDispatcher.

#33 - 10/31/2015 08:27 AM - Greg Shah

Feedback on the proposed workaround:

1. Why are there SwingEmulatedWindow instances that survive the reset of the client? ThinClient.getInstance() == null should never occur.
2. You say that "keyboard focus manager has not have an active window". Again, this is not a scenario that should ever occur. These changes don't solve that problem.
3. The removal of synchronized from WinKeyboardReader.keyPressed() and WinKeyboardReader.keyTyped() is not correct. There is usage of the keyDownEvents data structure. Are you sure that access to that data structure can only ever happen on a single thread?

#34 - 10/31/2015 10:04 AM - Sergey Ivanovskiy

Greg Shah wrote:

Feedback on the proposed workaround:

1. Why are there SwingEmulatedWindow instances that survive the reset of the client? ThinClient.getInstance() null should never occur.

I will try to explain the case ThinClient.getInstance() null but it happens with the focus listener.

2. You say that "keyboard focus manager has not have an active window". Again, this is not a scenario that should ever occur. These changes don't solve that problem.

These changes are indirectly set the active window, focus owner and focused window for the keyboard focus manager. Have you tried this fix and found that it doesn't work?

3. The removal of synchronized from WinKeyboardReader.keyPressed() and WinKeyboardReader.keyTyped() is not correct. There is usage of the keyDownEvents data structure. Are you sure that access to that data structure can only ever happen on a single thread?

These methods must be called only from the swing ui thread.

#35 - 10/31/2015 04:36 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Greg Shah wrote:

Feedback on the proposed workaround:

1. Why are there SwingEmulatedWindow instances that survive the reset of the client? ThinClient.getInstance() null should never occur.

I will try to explain the case ThinClient.getInstance() null but it happens with the focus listener.

To add on what Greg noted: when client is reset, please ensure these are cleaned up fully: windows (as widgets) and SwingGuiClient jframes (where SwingEmulatedWindows are parented). Note that we also use JWindow via GuiWindowContainer (for child windows, I think).

About the "ThinClient not null" cases: I think we need to ensure all driver windows are shutdown before resetting the ThinClient - I haven't checked, is TC.terminate now clearing all windows (including default)? As Greg noted, ThinClient instance should never be null (a SwingEmulatedWindow can never exist without a backing TC instance).

About SwingUtilities.invokeLater - please check the callpaths from where these are called; if there is a chance the main thread locks the driver via selectWindow, then this will cause a deadlock, as the swing UI thread might be waiting for that lock to be released. So you will need to enclose these in saveDrawingState and restoreDrawingState, so the swing UI thread can acquire the lock and do the work.

Finally, as all of these Swing cases share the same WinKeyboardReader, something else which should be checked: do you think it might be better to remove the WinKeyboardReader from the swing window/frame, before these are hidden or destroyed (and register it back when the window is made visible again)?

#36 - 11/01/2015 01:42 PM - Sergey Ivanovskiy

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

Greg Shah wrote:

Feedback on the proposed workaround:

1. Why are there SwingEmulatedWindow instances that survive the reset of the client? `ThinClient.getInstance()` null should never occur.

I will try to explain the case `ThinClient.getInstance()` null but it happens with the focus listener.

To add on what Greg noted: when client is reset, please ensure these are cleaned up fully: windows (as widgets) and `SwingGuiClient` jframes (where `SwingEmulatedWindows` are parented). Note that we also use `JWindow` via `GuiWindowContainer` (for child windows, I think).

About the "ThinClient not null" cases: I think we need to ensure all driver windows are shutdown before resetting the `ThinClient` - I haven't checked, is `TC.terminate` now clearing all windows (including default)? As Greg noted, `ThinClient` instance should never be null (a `SwingEmulatedWindow` can never exist without a backing `TC` instance).

About `SwingUtilities.invokeLater` - please check the callpaths from where these are called; if there is a chance the main thread locks the driver via `selectWindow`, then this will cause a deadlock, as the swing UI thread might be waiting for that lock to be released. So you will need to enclose these in `saveDrawingState` and `restoreDrawingState`, so the swing UI thread can be acquire the lock and do the work.

Constantin, please explain you thoughts more thoroughly, I have found `selectWindow` usages from swing UI dispatcher thread. Is it only thread that can call this driver method in the save manner. In this bug, there are no deadlocks. My only thoughts are it can be caused by incorrect usages of swing related api, for example creation, disposing, adding and removing listeners in non UI dispatcher thread can have an impact on the swing UI system.

Finally, as all of these Swing cases share the same `WinKeyboardReader`, something else which should be checked: do you think it might be better to remove the `WinKeyboardReader` from the swing window/frame, before these are hidden or destroyed (and register it back when the window is made visible again)?

Do you mean to have the unique instance of `WinKeyboardReader` and detach it from a window before it is hidden or destroyed and attach it again if a window becomes visible again? The current code should remove `WinKeyboardReader` on quit method of `SwingEmulatedWindow` and the instance of `WinKeyboardReader` is recreated on restart. I think that quit should be called from the swing UI dispatcher thread.

#37 - 11/01/2015 02:17 PM - Sergey Ivanovskiy

The problem is that the swing doesn't dispatch keys input. Please look at #31 note, at the moment of lost keys input the KeyboardFocusManager doesn't have an active window, a focused window and a focus owner. If we minimize the client window and then restore it, the keys input will be also restored and delivered to the keys listener (WinKeyboardReader).

#38 - 11/01/2015 02:27 PM - Sergey Ivanovskiy

In the note #36 quit should be called from the swing UI dispatcher thread. The dispose method of java.awt.Window can be used from non UI thread, but I think the listeners in spite of their thread safeness should be attached and detached from the UI thread.

#39 - 11/02/2015 03:38 PM - Sergey Ivanovskiy

Please, who can explain the resize method for EmulatedWindowState.java

```
public void resizeWindow(int width, int height)
{
    repaint();

    this.bitmap.setRows(height);
    this.bitmap.setColumns(width);
}
```

Is it done intentionally?

#40 - 11/02/2015 05:35 PM - Sergey Ivanovskiy

At this moment I have no any ideas why after the application restart, the swing client window doesn't get FocusGained event. If the breakpoint is inserted at WindowGuiImpl.show, after the process has been resumed, the swing client window gets FocusGained event and also it gets keys input. I think it should be initiated by the JFrame peer, but it doesn't happen. Working inside swing gui driver I can only force to request a focus using some swing api methods like posting corresponding events in the awt events queue.

#41 - 11/03/2015 03:06 AM - Sergey Ivanovskiy

I have tried to store the keyboard reader instance in the swing application context and to save the default window after restart as it is done now in the 2677a branch. But in that case the ThinClient instance is reset on ThinClient.terminate() and the focus listener from swing ui dispatcher thread can get null instance of ThinClient.getInstance(). Is it a correct way to work around the lost global focus issue?

#42 - 11/03/2015 03:17 AM - Sergey Ivanovskiy

- File workaround_20151102.txt added

Greg, please review the attached diff.

#43 - 11/03/2015 03:52 AM - Sergey Ivanovskiy

- File restart_workaround_20151103.txt added

This diff is ready to commit with comments.

#44 - 11/03/2015 08:59 AM - Greg Shah

I don't think this workaround is correct.

I still don't understand how we have a working window that is receiving input and focus events while the ThinClient is in the middle of initialization. That is not correct.

No residue of the prior run should remain at the time we restart. I think that means that we must go deeper with our cleanup processing. I also believe this means we do need to destroy the default-window too. If we do not, then the restarted application may have effects from the prior run as found in the default-window.

I think the cleanup described in notes 33 and 35 above is what needs to happen.

#45 - 11/03/2015 09:10 AM - Sergey Ivanovskiy

Greg Shah wrote:

I don't think this workaround is correct.

I still don't understand how we have a working window that is receiving input and focus events while the ThinClient is in the middle of initialization. That is not correct.

No residue of the prior run should remain at the time we restart. I think that means that we must go deeper with our cleanup processing. I also believe this means we do need to destroy the default-window too. If we do not, then the restarted application may have effects from the prior run as found in the default-window.

I think the cleanup described in notes 33 and 35 above is what needs to happen.

Ok, thus I should come back to the way to destroy all windows on the clients restart.

#46 - 11/03/2015 09:31 AM - Greg Shah

Yes. And it is important to remove listeners and other event processing instances so that we don't retain anything that can be invoked via events.

#47 - 11/03/2015 10:45 AM - Sergey Ivanovskiy

- *File lost_keys_input_fix_20151103_1.txt added*

Greg, please review the fix, all listeners are removed and windows are disposed during ThinClient.terminate(). I am ready to prove that it fixes this toolkit bug.

#48 - 11/03/2015 11:00 AM - Sergey Ivanovskiy

In this bug the keys input are always delivered to the swing KeyboardFocusManager, but the last one due to this toolkit bug loses an active window and hence doesn't dispatch keys input.

#49 - 11/03/2015 11:02 AM - Greg Shah

Sergey Ivanovskiy wrote:

In this bug the keys input are always delivered to the swing KeyboardFocusManager, but the last one due to this toolkit bug loses an active window and hence doesn't dispatch keys input.

I don't quite understand what you mean by this. Are you describing a 2nd bug? Does this bug exist after your change?

#50 - 11/03/2015 11:06 AM - Sergey Ivanovskiy

Greg Shah wrote:

Sergey Ivanovskiy wrote:

In this bug the keys input are always delivered to the swing KeyboardFocusManager, but the last one due to this toolkit bug loses an active window and hence doesn't dispatch keys input.

I don't quite understand what you mean by this. Are you describing a 2nd bug? Does this bug exist after your change?

No it fixes this bug. I have tried to do comments to my fix due to I am worry about this case.

#51 - 11/03/2015 11:07 AM - Greg Shah

Sergey Ivanovskiy wrote:

Greg, please review the fix, all listeners are removed and windows are disposed during ThinClient.terminate(). I am ready to prove that it fixes this toolkit bug.

I like this much better.

However, I am trying to understand the proposed change to SwingGuiDriver.

It seems that we could simply do this:

```
private static final SwingKeyboardReader keyReader = new WinKeyboardReader();
```

Is there a difference between this and your version?

Constantin: I don't think we really need the ContextLocal here. We probably don't even need to make it static since we never recreate a new instance of the driver itself. What do you think?

#52 - 11/03/2015 11:10 AM - Sergey Ivanovskiy

Agree, we can create the keyboard instance.

#53 - 11/03/2015 11:30 AM - Sergey Ivanovskiy

- *File lost_keys_input_fix_20151103_2.txt added*

Changed to create the keyboard reader as a final driver's instance field.
Greg, Constantin, please look at the attached diff.

#54 - 11/03/2015 11:32 AM - Greg Shah

I am fine with the change.

Constantin: what do you think?

#55 - 11/03/2015 12:42 PM - Constantin Asofiei

Greg Shah wrote:

I am fine with the change.

Constantin: what do you think?

Yes, I'm Ok with the changes. But I think we should call the WindowManager.resetWindow(); for GUI too, in ThinClient.terminate.

#56 - 11/03/2015 01:05 PM - Sergey Ivanovskiy

- *File lost_keys_input_fix_20151103_3.txt added*

Constantin Asofiei wrote:

Greg Shah wrote:

I am fine with the change.

Constantin: what do you think?

Yes, I'm Ok with the changes. But I think we should call the `WindowManager.resetWindow()`; for GUI too, in `ThinClient.terminate`.

Applied `resetWindow()`, please look at this diff. Ready for commit, but it should be checked with programs having more than two child windows.

#57 - 11/03/2015 01:07 PM - Sergey Ivanovskiy

- File `lost_keys_input_fix_20151103_3.txt` added

Forgot history comments.

#58 - 11/03/2015 01:47 PM - Sergey Ivanovskiy

- File `lost_keys_input_fix_20151103_4.txt` added

Greg, do you permit to commit the attached diff into 2677a?

#59 - 11/03/2015 02:02 PM - Sergey Ivanovskiy

I am looking at my system resource screen while starting the customer's p2j server, the cpu threads are playing together, one thread is busy, another is calm and then they are changed to calm and busy. I think one is waiting for the common resource owned by the other thread. The more cpu threads are nothing to do because the common resource is not shared.

#60 - 11/03/2015 02:14 PM - Sergey Ivanovskiy

I have applied this diff to the customer's GUI app and after select 454 testcases, the required screens are displayed, then click on "modify" button, the application is failed with this log at the end

```
java.lang.RuntimeException: No renderer is registered for id = 1
at com.goldencode.p2j.ui.client.gui.driver.GuiPrimitivesImpl.getWindowEmulator(GuiPrimitivesImpl.java:208)
at com.goldencode.p2j.ui.client.gui.driver.AbstractGuiDriver.enableEvents(AbstractGuiDriver.java:2161)
at com.goldencode.p2j.ui.client.WindowManager.demodalizeWindow(WindowManager.java:960)
at com.goldencode.p2j.ui.client.WindowManager.removeNoDestroy(WindowManager.java:659)
at com.goldencode.p2j.ui.client.WindowManager.remove(WindowManager.java:637)
at com.goldencode.p2j.ui.client.gui.ModalWindow$1.notifyWidgetDestroyed(ModalWindow.java:210)
at com.goldencode.p2j.ui.client.widget.AbstractWidget.destroy(AbstractWidget.java:1402)
at com.goldencode.p2j.ui.client.widget.AbstractContainer.destroy(AbstractContainer.java:865)
at com.goldencode.p2j.ui.client.widget.TitledWindow.destroy(TitledWindow.java:403)
at com.goldencode.p2j.ui.client.gui.ModalWindow.destroy(ModalWindow.java:654)
at com.goldencode.p2j.ui.client.WindowManager.clearWindowList(WindowManager.java:318)
at com.goldencode.p2j.ui.chui.ThinClient.terminate(ThinClient.java:2878)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:280)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:100)
at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:201)
at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:398)
```

at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:1)
at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:267)
For rev 111006 the result is the same, the client is failed if we press the "Modify" button with the same log at the end.

#61 - 11/03/2015 05:22 PM - Greg Shah

For rev 111006 the result is the same, the client is failed if we press the "Modify" button with the same log at the end.

To confirm: you are saying that your changes do not cause this issue?

#62 - 11/03/2015 05:24 PM - Greg Shah

Sergey Ivanovskiy wrote:

Greg, do you permit to commit the attached diff into 2677a?

If it does not cause any regression, then please do commit it to 2677a.

#63 - 11/04/2015 02:20 PM - Sergey Ivanovskiy

Greg Shah wrote:

Sergey Ivanovskiy wrote:

Greg, do you permit to commit the attached diff into 2677a?

If it does not cause any regression, then please do commit it to 2677a.

It is not ready, I have found the test case if ask-gui.p has the program /demo/simple_windows.p to run and then we sets another non existing program name to initiate restart, new window is added to the existing old windows opened by simple_windows.p. Thus clearWindowList() doesn't work. Now I am investigating the problem. The current diff is changed in order to set up a single instance of the driver's level keyboard focus manager.

#64 - 11/04/2015 02:21 PM - Sergey Ivanovskiy

- *File lost_keys_input_fix_20151104_1.txt added*

#65 - 11/04/2015 04:38 PM - Sergey Ivanovskiy

- *File lost_keys_input_fix_20151104_1.txt added*

Greg, please review committed revision 11008 branch 2677a. It clears the windows in the reverse order to their creations. It fixes the exception in note #60 and the restart testcase. Now if the customer's GUI testcase is failed, the client test application is restarted.

#66 - 11/04/2015 04:44 PM - Greg Shah

Code Review Task Branch 2677a Revision 11008

I'm OK with the changes.

Constantin: do you have any concerns?

Sergey: can we close this issue or are there any remaining items to address?

#67 - 11/04/2015 05:36 PM - Sergey Ivanovskiy

Sergey: can we close this issue or are there any remaining items to address?

I think we can close this issue.

#68 - 11/05/2015 05:35 AM - Constantin Asofiei

Greg Shah wrote:

Constantin: do you have any concerns?

No, the changes look good.

#69 - 11/05/2015 07:51 AM - Greg Shah

- *Status changed from New to Closed*

- *% Done changed from 0 to 100*

#70 - 11/13/2015 08:36 AM - Greg Shah

- Status changed from Closed to WIP

This has been regressed at some point in branch 2677a. The original problem is back.

#71 - 11/13/2015 09:42 AM - Sergey Ivanovskiy

The current version 11048 I used to test has the visibility window issue, the swing client window for ask-gui.p doesn't appear. Greg, can you run ask-gui.p? I need some help to find an entry point to debug the visibility issue.

Server side execute

```
def var prog as char format "x(74)".
def var again as logical init true.

current-window:width-chars = 109.

do while again:

  update prog label "Program Name" with frame fr-ask no-box side-labels size 105 by 1.
  run value(prog).

  message "Try another program?" update again format "Y/N".

end.
```

converted to

```
public void execute()
{
  externalProcedure(new Block()
  {
    character prog = new character("");

    logical again = new logical(true);

    public void init()
    {
      TransactionManager.register(prog, again);
    }

    public void body()
    {
      frAskFrame.openScope();
      currentWindow().unwrapSizeable().setWidthChars(new decimal(new integer(109)));

      doWhile("loopLabel0", new LogicalExpression(again), new Block()
      {
        public void body()
        {
          FrameElement[] elementList0 = new FrameElement[]
          {
            new Element(prog, frAskFrame.widgetProg())
          };

          frAskFrame.update(elementList0);

          ControlFlowOps.invoke((prog).toStringMessage());
          message("Try another program?", false, new AccessorWrapper(again), "Y/N");
        }
      });
    }
  });
}
```

From javadoc frAskFrame.update(elementList0);

```
* Copies data from the frame elements to each corresponding widget, then
* enables the list of widgets, shifts into input blocking mode via
* {@link #waitFor}, after the user's edits are complete the list of
* widgets are disabled and the edited data is copied from the screen
```

```
* buffer to the listed frame elements.  
* <p>  
* As part of the processing, the frame and each listed widget will be  
* marked as visible. The frame will be brought to the top of the Z-order  
* and refreshed.
```

it is not the driver's level issue, it is high level implementation issue.

#72 - 11/13/2015 09:52 AM - Greg Shah

I probably should have opened a new issue. The problem I'm referring to is in the web client. It is similar to the original Swing issue. The ask-gui.p actually does work in the web client except the initial rendering of the fill-in is broken. Just type the program name and hit enter. After that it will either work (because the program name was correct) or it will hang because the program name was wrong.

The problem you are referring to in the Swing client is probably something different. I just looked at that. We are in:

```
Thread [main] (Suspended)  
  waiting for: TypeAhead (id=119)  
  Object.wait(long) line: not available [native method]  
  TypeAhead.getKeyStroke(int, boolean) line: 354  
  ThinClient.waitForEvent(int, boolean, boolean, boolean) line: 12866  
  ThinClient.waitForWorker(EventList, int, int, ScreenBuffer, boolean, boolean, boolean, BlockingOperation)  
line: 10858  
  ThinClient.waitFor(EventList, int, int, ScreenBuffer, BlockingOperation, boolean) line: 10441  
  ThinClient.promptFor(int, int[], ScreenBuffer, int, EventList, boolean, int) line: 9139  
  NativeMethodAccessorImpl.invoke0(Method, Object, Object[]) line: not available [native method]  
  NativeMethodAccessorImpl.invoke(Object, Object[]) line: 62  
  DelegatingMethodAccessorImpl.invoke(Object, Object[]) line: 43  
  Method.invoke(Object, Object...) line: 497  
  MethodInvoker.invoke(Object[]) line: 76  
  Dispatcher.processInbound(InboundRequest, boolean, NetResource) line: 705  
  Conversation.block() line: 319  
  Conversation.waitForMessage(int) line: 257  
  Queue.transactImpl(Message, int) line: 1128  
  Queue.transact(Message, int) line: 585  
  DirectSession(BaseSession).transact(Message, int) line: 223  
  HighLevelObject.transact(RoutingKey, Object[]) line: 163  
  RemoteObject$RemoteAccess.invokeCore(Object, Method, Object[]) line: 1425  
  RemoteObject$RemoteAccess(InvocationStub).invoke(Object, Method, Object[]) line: 97  
  $Proxy4.standardEntry(ClientParameters) line: not available  
  ClientCore.start(BootstrapConfig, boolean, boolean) line: 277  
  ClientCore.start(BootstrapConfig, boolean) line: 100  
  ClientDriver.start(BootstrapConfig) line: 201  
  ClientDriver(CommonDriver).process(String[]) line: 398  
  ClientDriver.process(String[]) line: 95  
  ClientDriver.main(String[]) line: 267
```

Everything looks OK and there are no uncaught exceptions before this point. I think the problem is that the enable() call in the ThinClient.promptFor() is not making the window visible for some reason. I can't explain why this happens on Swing and not in the web client. Though the web client also does have the unresponsive fill-in issue. The web client also cannot be moved (mouse on titlebar) until after the prompt-for returns. So this may also be related to the enable().

#73 - 11/13/2015 10:03 AM - Sergey Ivanovskiy

line 9431 the variable allFramesVisible is false and we are in ThinClient.enable in ThinClient.viewWorker(...) What can we do?

```
final TopLevelWindow window = WindowManager.findWindow(rfcfg.windowID);
if (allFramesVisible      &&
    window != null        &&
    !window.parentOrSelfHidden() &&
    (!window.config().realized || !window.isVisible()))
{
    independentEventDrawingBracket(window, new Runnable()
    {
        public void run()
        {
            window.show();
        }
    });
}
```

#74 - 11/13/2015 10:15 AM - Sergey Ivanovskiy

Should the default window become visible if it is realized?

```
public static synchronized <O extends OutputManager<?>> Window<O> getDefaultWindow()
{
    Window<?> window = defaultWnd.get();

    if (window == null)
    {
        OutputManager<?> om = OutputManager.instance();
        window = om.getFactory().createWindow(WidgetId.DEFAULT_WINDOW_ID);

        om.activateWindow(window);

        defaultWnd.set(window);
        currentWnd.set(window);

        // add it to the window manager
        WindowManager.addWindow(window);

        // ensure the window is targeted by the output manager
        om.switchOutput(null);
    }

    return (Window<O>) window;
}
```


#75 - 11/13/2015 10:17 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

Should the default window become visible if it is realized?
[...]

No. DEFAULT-WINDOW is not visible/not hidden implicitly. It gets visible only when something is drawn on them. I'll take a look at ask-gui.p and see why it fails.

#76 - 11/13/2015 10:31 AM - Constantin Asofiei

OK. The reason why ask-gui.p fails is because currently in GUI the UPDATE, SET and PROMPT-FOR don't set the frame's VISIBLE attribute to true on client-side, before performing the VIEW. This is related to the changes in [#2809](#).

Sergey, if it's OK, I'm taking this.

#77 - 11/13/2015 10:36 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

OK. The reason why ask-gui.p fails is because currently in GUI the UPDATE, SET and PROMPT-FOR don't set the frame's VISIBLE attribute to true on client-side, before performing the VIEW. This is related to the changes in [#2809](#).

Sergey, if it's OK, I'm taking this.

Yes, of course.

#78 - 11/13/2015 12:07 PM - Constantin Asofiei

The fix for the problem in note 71/72 is in 2677a rev 11049

#79 - 11/13/2015 12:27 PM - Sergey Ivanovskiy

Checked the swing client the ask-gui.p window appears but the modal dialog "Try another program?" doesn't appear. It is interesting the chui client is influenced by this bug.

#80 - 11/13/2015 12:52 PM - Sergey Ivanovskiy

Debugging the code WindowGuiImpl.tinyInput found that again we fall down to

```
// check if the window is hidden; if so, make it visible
final TopLevelWindow window = WindowManager.findWindow(rfcfg.windowID);
```

```

if (allFramesVisible      &&
    window != null        &&
    !window.parentOrSelfHidden() &&
    (!window.config().realized || !window.isVisible()))
{
    independentEventDrawingBracket(window, new Runnable()
    {
        public void run()
        {
            window.show();
        }
    });
}

```

but now `window.config()` returns null, where `window` is `FrameWindowGuiImpl`.

#81 - 11/13/2015 12:57 PM - Sergey Ivanovskiy

Hynek, Constantin please help what implementation of

```

/**
 * Get externalizable widget config which can be used to restore widget
 * state after via network.
 *
 * Note that only very limited set of widgets actually have config.
 * By default widget has no config.
 *
 * @param <W>
 *         Config type.
 *
 * @return Reference to widget config.
 */
@Override
public <W extends WidgetConfig> W config()
{
    return null;
}

```

should be for `FrameWindowGuiImpl`?

#82 - 11/13/2015 01:11 PM - Sergey Ivanovskiy

- File 11050.txt added

Decided to commit revision 11050 because config() hasn't been implemented for FrameWindowGuiImpl.

#83 - 11/13/2015 01:35 PM - Sergey Ivanovskiy

Now for the swing client ask-gui.p displays "Try another program?" but if we give ask-gui.p as an input program this message doesn't appear. Continue the input with hello.p, the "Try another program?" dialog appears, type N, the result the second dialog "Try another program?" corresponding to ask-gui.p input appears. Greg, what do you think it is correct? I think this behavior has been before all recent changes. The second question is for the web client what is the behavior we expect if the input program is incorrect? Now it falls down to login page. Is it expected?

#84 - 11/13/2015 02:20 PM - Greg Shah

Yes, ask-gui.p is working properly now.

However, I would like Constantin or Hynek to comment on rev 11050. The change is safe enough to leave there, but the fact that we didn't expect it to be null suggests that something else is wrong.

#85 - 11/13/2015 02:33 PM - Hynek Cihlar

Greg Shah wrote:

Yes, ask-gui.p is working properly now.

However, I would like Constantin or Hynek to comment on rev 11050. The change is safe enough to leave there, but the fact that we didn't expect it to be null suggests that something else is wrong.

It is not expected for FrameWindowGuiImpl to have an associated config instance because it doesn't synchronize its state with server.

The change in 11050 is ok. An improvement would be to have isRealized() method which would forward to config().realized for widgets with valid config and to ModalWindow.realized for ModalWindow and descendants.

#86 - 11/13/2015 03:01 PM - Hynek Cihlar

Hynek Cihlar wrote:

Greg Shah wrote:

Yes, ask-gui.p is working properly now.

However, I would like Constantin or Hynek to comment on rev 11050. The change is safe enough to leave there, but the fact that we didn't expect it to be null suggests that something else is wrong.

It is not expected for `FrameWindowGuiImpl` to have an associated config instance because it doesn't synchronize its state with server.

The change in 11050 is ok. An improvement would be to have `isRealized()` method which would forward to `config().realized` for widgets with valid config and to `ModalWindow.realized` for `ModalWindow` and descendants.

BTW, the condition `!window.config().realized || !window.isVisible()` is interesting. Is it even possible for the window to be unrealized when visible? The realized check seems to be redundant.

#87 - 11/13/2015 03:26 PM - Greg Shah

After testing `ask-gui.p` in the web client, it dies as soon as the UPDATE exits. The Swing client works fine. In the web client log file is this:

```
Nov 13, 2015 3:22:01 PM com.goldencode.p2j.ui.client.widget.AbstractContainer detach  
WARNING: Cannot detach, widget BorderedPanelGuiImpl:null is not attached to container NativeScrollContainer:nu  
ll.
```

#88 - 11/14/2015 09:38 AM - Greg Shah

BTW, the condition `!window.config().realized || !window.isVisible()` is interesting. Is it even possible for the window to be unrealized when visible? The realized check seems to be redundant.

You are probably correct here. The opposite is true (we can be realized and not visible). But it doesn't seem right to be unrealized and visible.

#89 - 11/14/2015 09:39 AM - Greg Shah

An improvement would be to have `isRealized()` method which would forward to `config().realized` for widgets with valid config and to `ModalWindow.realized` for `ModalWindow` and descendants.

Please go ahead with this change. Leaving the null check just looks like it is a patch and not a designed idea.

#90 - 11/16/2015 05:58 AM - Sergey Ivanovskiy

After testing ask-gui.p in the web client, it dies as soon as the UPDATE exits. The Swing client works fine. In the web client log file is this:

Greg, the current version works as it is described below:

- 1) Login the web client.
- 2) Ask-gui.p test program is displayed.
- 3) Print a existing program, hello.p and press ENTER
- 4) hello.p is done and "Try another program?" Yes/No dialog appears.
- 5) If print Y, then the focus is on the input program field and if print N, then the dialog is closed and the procedure complete message is displayed.
- 6) Print a nonexistent program.
- 7) The current web session is closed and the client is redirected to the login web page.

Is it correct?

#91 - 11/16/2015 06:00 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

After testing ask-gui.p in the web client, it dies as soon as the UPDATE exits. The Swing client works fine. In the web client log file is this:

Greg, the current version works as it is described below:

- 1) Login the web client.
- 2) Ask-gui.p test program is displayed.
- 3) Print a existing program, hello.p and press ENTER
- 4) hello.p is done and "Try another program?" Yes/No dialog appears.
- 5) If print Y, then the focus is on the input program field and if print N, then the dialog is closed and the procedure complete message is displayed.
- 6) Print a nonexistent program.
- 7) The current web session is closed and the client is redirected to the web page.

Is it correct?

I think step 7 is correct: when a nonexistent program/procedure is passed to a RUN statement, a STOP condition is raised in 4GL. If not caught (via a ON STOP clause) it will restart the client.

#92 - 11/16/2015 06:29 AM - Sergey Ivanovskiy

Greg Shah wrote:

An improvement would be to have `isRealized()` method which would forward to `config().realized` for widgets with valid config and to `ModalWindow.realized` for `ModalWindow` and descendants.

Please go ahead with this change. Leaving the null check just looks like it is a patch and not a designed idea.

Greg, Constantin, Hynek, we can add boolean `isRealized()` to `Widget` interface, the last one declares the similar method `boolean isVisible()`. Otherwise, we can add new protected method to `AbstractWidget` class, it is possible since `ModalWindow` is in the class hierarchy with `AbstractWidget` as a root. What does it mean for a particular widget to be realized or not and is it different from boolean `isDisplayed()` of `Widget`? For a window I would interpret it as to has a physical instance registered by the gui driver. Is it a correct meaning?

#93 - 11/16/2015 09:05 AM - Greg Shah

Constantin Asofiei wrote:

Sergey Ivanovskiy wrote:

After testing `ask-gui.p` in the web client, it dies as soon as the `UPDATE` exits. The Swing client works fine. In the web client log file is this:

Greg, the current version works as it is described below:

- 1) Login the web client.
 - 2) `Ask-gui.p` test program is displayed.
 - 3) Print a existing program, `hello.p` and press `ENTER`
 - 4) `hello.p` is done and "Try another program?" Yes/No dialog appears.
 - 5) If print `Y`, then the focus is on the input program field and if print `N`, then the dialog is closed and the procedure complete message is displayed.
 - 6) Print a nonexistent program.
 - 7) The current web session is closed and the client is redirected to the web page.
- Is it correct?

I think step 7 is correct: when a nonexistent program/procedure is passed to a `RUN` statement, a `STOP` condition is raised in 4GL. If not caught (via a `ON STOP` clause) it will restart the client.

In the Swing client, we just loop in `ClientCore.start()`. As long as running (as returned from `standardEntry()`) is not false, then we will just reset the client WITHOUT exiting from the client process.

In the web client at step 7, we are no longer connected to the client's Jetty. We are put back to the operating system login screen (P2J server's Jetty). This is definitely incorrect.

All other prior steps are OK.

#94 - 11/16/2015 09:27 AM - Greg Shah

What does it mean for a particular widget to be realized or not and is it different from boolean `isDisplayed()` of `Widget`? For a window I would interpret it as to has a physical instance registered by the gui driver. Is it a correct meaning?

My best understanding is that in the 4GL, "realized" means that the widget has been through the layout/sizing algorithm and has been placed in a location relative to its container, with a known size. Sometimes that may have been an explicit location/size specified by the programmer. Sometimes it is implicitly chosen by the arcane rules hidden inside the 4GL.

Just because something is realized, does NOT mean it is visible. However, the widget must be realized before being made visible. As far as I understand it, once realized a widget can never be made unrealized, though I wonder if that is correct for dynamic widgets (e.g. disconnecting a widget from its container may make something unrealized, I haven't checked it).

Widgets can be made non-visible but that doesn't make the widget unrealized.

#95 - 11/16/2015 10:40 AM - Sergey Ivanovskiy

- *File realized_1.txt added*

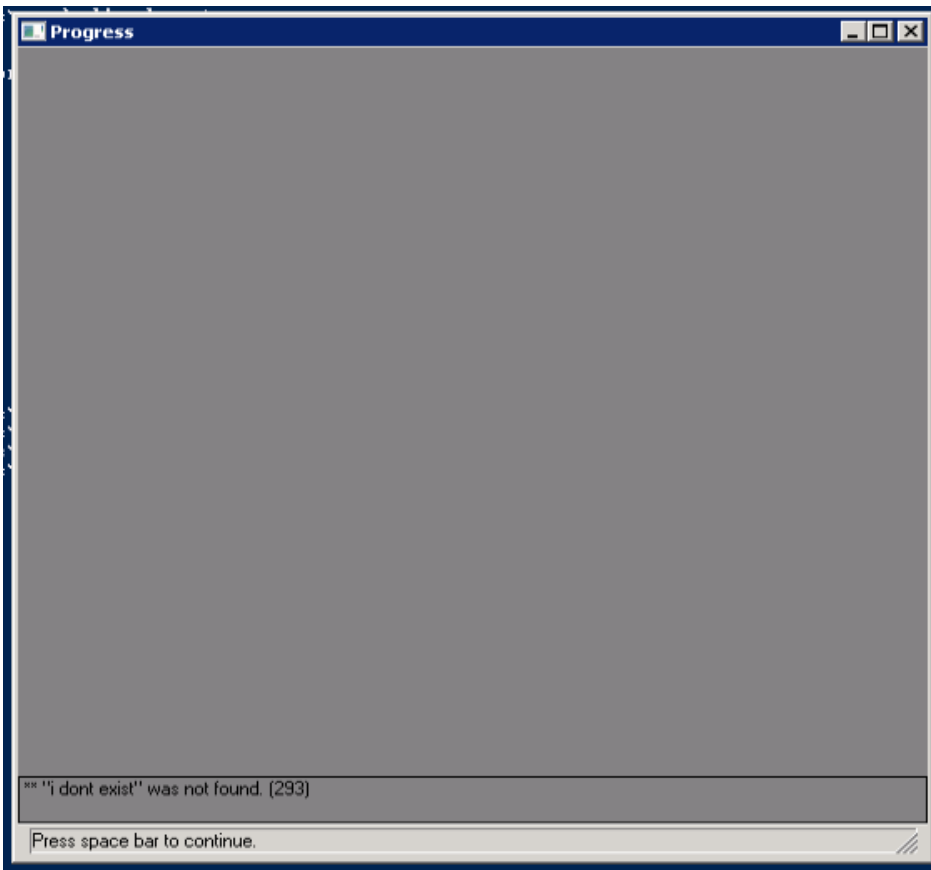
Greg, please review the changes according to #89. The default implementation `isRealized()` returns `config() != null ? config().realized : false`. `ModalWindow` already implements `isRealized()`.

#96 - 11/16/2015 12:32 PM - Greg Shah

- *File ask-gui.p_wrong_program_name_in_progress_20151116.png added*

Based on testing in rev 11052, even the Swing client is acting improperly. The Swing client seems to flash the error message and then restart without a pause.

But in the 4GL, the `ask-gui.p` will definitely pause before restarting:



#97 - 11/16/2015 12:46 PM - Sergey Ivanovskiy

Greg Shah wrote:

In the web client at step 7, we are no longer connected to the client's Jetty. We are put back to the operating system login screen (P2J server's Jetty). This is definitely incorrect.

The server throws the following exception if the web client tries to re-establish a connection by session = sessMgr.connectDirect(new InterruptedExceptionHandler(), null); (ClientCore)

```
(Incoming.run():WARNING) {Incoming SSL Connector} initialization failure
javax.net.ssl.SSLEException: Connection has been shutdown: javax.net.ssl.SSLHandshakeException: Received fatal
alert: certificate_unknown
    at sun.security.ssl.SSLSocketImpl.checkEOF(SSLSocketImpl.java:1529)
    at sun.security.ssl.SSLSocketImpl.checkWrite(SSLSocketImpl.java:1541)
    at sun.security.ssl.AppOutputStream.write(AppOutputStream.java:71)
    at java.io.ObjectOutputStream$BlockDataOutputStream.drain(ObjectOutputStream.java:1877)
    at java.io.ObjectOutputStream$BlockDataOutputStream.setBlockDataMode(ObjectOutputStream.java:1786)
    at java.io.ObjectOutputStream.<init>(ObjectOutputStream.java:247)
    at com.goldencode.p2j.net.NetSocket.<init>(NetSocket.java:79)
    at com.goldencode.p2j.net.RouterSessionManager$Incoming.run(RouterSessionManager.java:1389)
    at java.lang.Thread.run(Thread.java:745)
Caused by: javax.net.ssl.SSLHandshakeException: Received fatal alert: certificate_unknown
    at sun.security.ssl.Alerts.getSSLEException(Alerts.java:192)
    at sun.security.ssl.Alerts.getSSLEException(Alerts.java:154)
    at sun.security.ssl.SSLSocketImpl.recvAlert(SSLSocketImpl.java:2011)
    at sun.security.ssl.SSLSocketImpl.readRecord(SSLSocketImpl.java:1113)
    at sun.security.ssl.SSLSocketImpl.performInitialHandshake(SSLSocketImpl.java:1363)
    at sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:1391)
    at sun.security.ssl.SSLSocketImpl.getSession(SSLSocketImpl.java:2225)
    at com.goldencode.p2j.net.NetSocket.<init>(NetSocket.java:75)
    at com.goldencode.p2j.net.RouterSessionManager$Incoming.run(RouterSessionManager.java:1389)
    at java.lang.Thread.run(Thread.java:745)
```

The server throws the following hidden null pointer during the step 1) "login the web client"

```
(Dispatcher.processInbound():SEVERE) {00000013:00000040:xo0c0572a9n7m1xd} Unexpected throwable.
java.lang.reflect.InvocationTargetException
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:497)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:705)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
    at java.lang.Thread.run(Thread.java:745)
Caused by: java.lang.NullPointerException
    at com.goldencode.p2j.main.SpawnerImpl.getTemporaryClient(SpawnerImpl.java:108)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:497)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:705)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
    at java.lang.Thread.run(Thread.java:745)
```

This exception is wrapped by SilentUnwindException and becomes hidden and looks like

```
Exception in thread "Reader [0000000F:99029fdg3g21flxi]" com.goldencode.p2j.net.SilentUnwindException: Connect
ion ended abnormally
    at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1420)
    at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:97)
    at com.sun.proxy.$Proxy4.isChui(Unknown Source)
    at com.goldencode.p2j.ui.LogicalTerminal.isChui(LogicalTerminal.java:8107)
    at com.goldencode.p2j.ui.LogicalTerminal.init(LogicalTerminal.java:962)
    at com.goldencode.p2j.security.ContextLocal.initializeValue(ContextLocal.java:566)
    at com.goldencode.p2j.security.ContextLocal.get(ContextLocal.java:409)
    at com.goldencode.p2j.security.ContextLocal.get(ContextLocal.java:367)
    at com.goldencode.p2j.ui.LogicalTerminal.locate(LogicalTerminal.java:10073)
    at com.goldencode.p2j.ui.LogicalTerminal.access$400(LogicalTerminal.java:727)
    at com.goldencode.p2j.ui.LogicalTerminal$3.createScopeable(LogicalTerminal.java:11034)
    at com.goldencode.p2j.util.TransactionManager$ContextContainer.obtain(TransactionManager.java:5794)
    at com.goldencode.p2j.util.TransactionManager.abortProcessing(TransactionManager.java:3754)
    at com.goldencode.p2j.net.SessionManager.endSession(SessionManager.java:1120)
```

```
at com.goldencode.p2j.net.SessionManager.queueStopped(SessionManager.java:1587)
at com.goldencode.p2j.net.Queue.stop(Queue.java:410)
at com.goldencode.p2j.net.Protocol$Reader.run(Protocol.java:415)
at java.lang.Thread.run(Thread.java:745)
```

#98 - 11/17/2015 01:06 AM - Sergey Ivanovskiy

- File `secure_connection.txt` added

The configuration that is used to restore the client session is

```
{WEB={REFERRER={URL=https://127.0.0.1:7443/gui}}, SERVER={SPAWNER={UUID=38a96f38-f2f8-40f4-8771-c148a80080c0}}
, NET={CONNECTION={SECURE=true}, SERVER={SECURE_PORT=7448, HOST=localhost}, QUEUE={START_THREAD=false, CONVERSATION=true}}, CLIENT={WEB={PORT=7449, SOCKETTIMEOUT=-1, HOST=localhost, WATCHDOGTIMEOUT=-1}, DRIVER={TYPE=gui_web}, GUI={DESKTOPWIDTH=1024, TASKBAR=true, DESKTOPHEIGHT=768}}
```

The temporary client's configuration that is used to establish the client session is

```
{SERVER={SPAWNER={UUID=38a96f38-f2f8-40f4-8771-c148a80080c0}}, SECURITY={AUTHENTICATION={TYPE=program}, TRUST_MGR={DISABLE=true}, CERTIFICATE={VALIDATE=false}}, NET={CONNECTION={SECURE=true}, SERVER={SECURE_PORT=7448, HOST=localhost}, QUEUE={START_THREAD=false, CONVERSATION=true}}, ACCESS={PASSWORD={USER=5e9j`02D!9+5S<03}, SUBJECT={ID=8o5PF9Vrwp8246a9}}, CLIENT={WEB={PORT=7449, SOCKETTIMEOUT=-1, HOST=localhost, WATCHDOGTIMEOUT=-1}}
```

The difference is `TRUST_MGR={DISABLE=true}, CERTIFICATE={VALIDATE=false}`, and given `ACCESS={PASSWORD={USER=5e9j`02D!9+5S<03}, SUBJECT={ID=8o5PF9Vrwp8246a9}}`.

Not sure that the diff is correct solution for this session security issue. But an insecure port is configured for the swing client.

#99 - 11/17/2015 01:20 AM - Sergey Ivanovskiy

Please help to find the documentation how to setup client-server session correctly. I don't know the security bootstrap configuration given by `p2j_runtime_installation_configuration_and_administration_guide.pdf` is up to date?

#100 - 11/17/2015 02:18 PM - Sergey Ivanovskiy

Greg, please look at the log produced during the web login process

```
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=64m; support was removed in 8.0
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000007:standard} TLS connection without certificate.
```

TLS connection without certificate AUTHTYPE = AUTH_REQ_PROGRAM

```

[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000007:standard} Getting AUTHTYPE
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000007:standard} Got AUTHTYPE: 2 userName: null
[11/17/2015 21:33:33 MSK] (SecurityManager:FINE) {00000000:00000007:standard} hook class is com.goldencode.p2j
.security.GuestAccess
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000007:standard} Sending AUTHMODEREQ: 1 (action
= 1)
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000007:standard} Sent AUTHMODEREQ
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000007:standard} Getting AUTHDATA
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000007:standard} Getting AUTHDATA authSize=36
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000007:standard} Got AUTHDATA
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000007:standard} Received user ID <3M835032oLq5o
wnS>
[11/17/2015 21:33:33 MSK] (SecurityManager:FINEST) {00000000:00000007:standard} Received password <960(5(#1Fq%
75XC7>

```

Temporary account is used to start the secure session

```

[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000001:00000007:3m835032oLq5owns} Search open: resId 0, i
nstance logon, mode 0, handle 70
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000001:00000007:3m835032oLq5owns} Search next: handle 70,
rights {true}
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000001:00000007:3m835032oLq5owns} Search done: handle 70,
decision true, cache false
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000007:standard} Sending AUTHRESULT0 (action = 0
)
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000007:standard} Sent AUTHRESULT
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000007:standard} Sending LOCALUSERID: 3M835032oL
q5ownS
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search open: resId 0, instance
admin, mode 0, handle 71
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search next: handle 71, rights
{true}
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search done: handle 71, decisio
n true, cache false
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search open: resId 0, instance
admin, mode 0, handle 72
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search next: handle 72, rights
{true}
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search done: handle 72, decisio
n true, cache false
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search open: resId 0, instance
admin, mode 0, handle 73
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search next: handle 73, rights
{true}
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search done: handle 73, decisio
n true, cache false
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search open: resId 0, instance
admin, mode 0, handle 74
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search next: handle 74, rights
{true}
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search done: handle 74, decisio
n true, cache false
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search open: resId 0, instance
admin, mode 0, handle 75
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search next: handle 75, rights
{true}
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search done: handle 75, decisio
n true, cache false
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search open: resId 0, instance
admin, mode 0, handle 76
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search next: handle 76, rights
{true}
[11/17/2015 21:33:33 MSK] (SecurityManager:FINER) {00000000:00000006:standard} Search done: handle 76, decisio
n true, cache false

```

The temporary session should be closed and the client initiates new one. AUTHTYPE = AUTH_REQ_USER, if the mode is AUTH_MODE_CUSTOM, the server should have sent and the client reads AUTHCUSTOM packet which delivers the class name and parameters of the custom authentication hook; this hook class is run (which can display any UI and interact with the user OR it may access any device/facility such as a smart card); the result is a userid and password

The key point `SecurityManager.authenticateLocal(NetSocket socket)` uses `SecurityCache sc = getCache();`

```
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} TLS connection without certificate.
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Getting AUTHTYPE
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Got AUTHTYPE: 0 userName: null
[11/17/2015 21:33:34 MSK] (SecurityManager:FINE) {00000000:0000000B:standard} hook class is com.goldencode.p2j.security.GuestAccess
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Sending AUTHMODEREQ: 4 (action = 1)
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Sent AUTHMODEREQ
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Sending AUTHCUSTOM
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Sent AUTHCUSTOM
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Getting AUTHDATA
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Getting AUTHDATA authSize=9
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Got AUTHDATA
```

Gets bogus user. Where is the corresponding code within `ClientCore` that sends bogus? Sure that it comes from the server using `AUTH_MODE_CUSTOM`. The directory contains `auth-plugins`

```
<node class="container" name="auth-plugins">
  <node class="container" name="guest_login">
    <node class="string" name="classname">
      <node-attribute name="value" value="com.goldencode.p2j.security.GuestAccess"/>
    </node>
    <node class="string" name="description">
      <node-attribute name="value" value="Non-Interactive Guest Auto-Login"/>
    </node>
    <node class="string" name="option">
      <node-attribute name="value" value="bogus"/>
    </node>
  </node>
</node>
```

the server sends `GuestAccess` to the client with option field "bogus" in order the client gets an access to the server.

```
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Received user ID <bogus>
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000002:0000000B:bogus} Search open: resId 0, instance log on, mode 0, handle 77
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000002:0000000B:bogus} Search next: handle 77, rights {true}
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000002:0000000B:bogus} Search done: handle 77, decision true, cache false
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Sending AUTHRESULT0 (action = 0)
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Sent AUTHRESULT
[11/17/2015 21:33:34 MSK] (SecurityManager:FINER) {00000000:0000000B:standard} Sending LOCALUSERID: bogus
[11/17/2015 21:33:34 MSK] (LogicalTerminal:WARNING) No client parameters are set: calling the client-side to determine the driver type is expensive!
[11/17/2015 21:33:35 MSK] (com.goldencode.p2j.ui.FontTable:SEVERE) Could not load the legacy font metrics from file font-metrics-ext.xml
[11/17/2015 21:33:35 MSK] (com.goldencode.p2j.ui.FontTable:SEVERE) Could not load the legacy text metrics from file text-metrics.xml
[11/17/2015 21:33:35 MSK] (com.goldencode.p2j.ui.FontTable:SEVERE) Could not load the custom font [Segue UI] from file fonts/segoeuibd.ttf
[11/17/2015 21:33:36 MSK] (LogicalTerminal:WARNING) No client parameters are set: calling the client-side to determine the driver type is expensive!
[11/17/2015 21:33:36 MSK] (LogicalTerminal:WARNING) No client parameters are set: calling the client-side to determine the driver type is expensive!
[11/17/2015 21:33:36 MSK] (LogicalTerminal:WARNING) No client parameters are set: calling the client-side to determine the driver type is expensive!
[11/17/2015 21:33:37 MSK] (com.goldencode.p2j.ui.FontTable:SEVERE) Text [Program Name: ] has no legacy metrics with font [ms sans serif,8,false,false,false].
[11/17/2015 21:33:37 MSK] (com.goldencode.p2j.ui.FontTable:SEVERE) Text [Program Name] has no legacy metrics with font [ms sans serif,8,false,false,false].
[11/17/2015 21:33:37 MSK] (com.goldencode.p2j.ui.FontTable:SEVERE) Text [: ] has no legacy metrics with font [ms sans serif,8,false,false,false].
```

For some reason `Reader` thread is not closed gently, exception is thrown in the final block

```
Exception in thread "Reader [00000001:3m835032olq5owns]" com.goldencode.p2j.net.SilentUnwindException: Connection ended abnormally
    at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1420)
    at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:97)
    at com.sun.proxy.$Proxy4.isChui(Unknown Source)
    at com.goldencode.p2j.ui.LogicalTerminal.isChui(LogicalTerminal.java:8107)
    at com.goldencode.p2j.ui.LogicalTerminal.init(LogicalTerminal.java:962)
    at com.goldencode.p2j.security.ContextLocal.initializeValue(ContextLocal.java:566)
    at com.goldencode.p2j.security.ContextLocal.get(ContextLocal.java:409)
    at com.goldencode.p2j.security.ContextLocal.get(ContextLocal.java:367)
    at com.goldencode.p2j.ui.LogicalTerminal.locate(LogicalTerminal.java:10073)
    at com.goldencode.p2j.ui.LogicalTerminal.access$400(LogicalTerminal.java:727)
    at com.goldencode.p2j.ui.LogicalTerminal$3.createScopeable(LogicalTerminal.java:11034)
    at com.goldencode.p2j.util.TransactionManager$ContextContainer.obtain(TransactionManager.java:5794)
    at com.goldencode.p2j.util.TransactionManager.abortProcessing(TransactionManager.java:3754)
    at com.goldencode.p2j.net.SessionManager.endSession(SessionManager.java:1120)
    at com.goldencode.p2j.net.SessionManager.queueStopped(SessionManager.java:1587)
    at com.goldencode.p2j.net.Queue.stop(Queue.java:410)
    at com.goldencode.p2j.net.Protocol$Reader.run(Protocol.java:415)
    at java.lang.Thread.run(Thread.java:745)
```

#101 - 11/17/2015 03:02 PM - Greg Shah

Sergey Ivanovskiy wrote:

Greg, please review the changes according to #89. The default implementation isRealized() returns config() != null ? config().realized : false. ModalWindow already implements isRealized().

The following is feedback from Hynek:

"no major concerns. Returning false from AbstractWidget.isRealized() when config is null is probably fine. AbstractWidget.isRealized() could be shortened to return widgetConfig != null && widgetConfig.realized;"

Sergey: please make the recommended change and check it in to 2677a if you can do this quickly. We are about to start testing again soon but I'd like to get this in.

#102 - 11/17/2015 03:07 PM - Sergey Ivanovskiy

Ok, can do it quickly.

#103 - 11/17/2015 03:27 PM - Sergey Ivanovskiy

- *File 11056.txt added*

Committed revision 11056 is checked for ask-gui.p

#104 - 11/17/2015 05:56 PM - Greg Shah

Please help to find the documentation how to setup client-server session correctly.

A good summary can be found in the `p2j_developer_guide.pdf` in the section titled "Establishing a Session with the P2J Server".

I don't know the security bootstrap configuration given by `p2j_runtime_installation_configuration_and_administration_guide.pdf` is up to date?

I think the parts that provide a reference to the bootstrap config values are mostly correct. It is possible we have added/removed/changed some usage in the last 3 years. I have not reviewed it in a while, but the reference information should be pretty good.

In regard to notes 97, 98 and 100:

I think our primary problem is that we are trying to reuse a `TemporaryClient` session that is no longer valid. It can't be valid at this point. But I don't think it needs to be used again anyway. We should be connecting back to the server using the same authentication credentials (bogus) we used in to establish the primary session in the first place.

Please see page 3 of the attached PDF. It shows the "spawning flow" for the web client. When we are "resetting" the client we want to avoid steps 1 through 8. Those are the server-driven steps to bootstrap and spawn the client process in the first place. In the reset case, we want to reuse the current process that is already spawned. We just need to re-establish the session and start fresh in the client.

#105 - 11/17/2015 05:57 PM - Greg Shah

- *File p2j_runtime_architecture_20141207.pdf added*

#106 - 11/18/2015 12:36 PM - Sergey Ivanovskiy

Greg Shah wrote:

Please help to find the documentation how to setup client-server session correctly.

A good summary can be found in the [p2j_developer_guide.pdf](#) in the section titled "Establishing a Session with the P2J Server".

I don't know the security bootstrap configuration given by [p2j_runtime_installation_configuration_and_administration_guide.pdf](#) is up to date?

I think the parts that provide a reference to the bootstrap config values are mostly correct. It is possible we have added/removed/changed some usage in the last 3 years. I have not reviewed it in a while, but the reference information should be pretty good.

In regard to notes 97, 98 and 100:

I think our primary problem is that we are trying to reuse a `TemporaryClient` session that is no longer valid. It can't be valid at this point. But I don't think it needs to be used again anyway. We should be connecting back to the server using the same authentication credentials (bogus) we used in to establish the primary session in the first place.

Please see page 3 of the attached PDF. It shows the "spawning flow" for the web client. When we are "resetting" the client we want to avoid steps 1 through 8. Those are the server-driven steps to bootstrap and spawn the client process in the first place. In the reset case, we want to reuse the current process that is already spawned. We just need to re-establish the session and start fresh in the client.

Greg, thanks for this spawner presentation, but there exist the temporary user accounts used to create session by programmatic authentication ([com/goldencode/p2j/security/package.html](#)).

What is the purpose of this step 7b if the client session can be created directly using another method or can't?

In `ClientCore` `SecurityManager` clears its cache after restart. Then the client is tried to open a session

```
session = sessMgr.connectDirect(new InterruptedExceptionHandler(), null);
```

using the bootstrap configuration provided by spawner

```
WEB:REFERER:URL=https://127.0.0.1:7443/gui
SERVER:SPAWNER:UUID=0fada8e4-c9a3-496c-8d81-5efe6b97913a
NET:CONNECTION:SECURE=true
NET:SERVER:SECURE_PORT=3334
NET:SERVER:HOST=localhost
NET:QUEUE:START_THREAD=false
NET:QUEUE:CONVERSATION=true
CLIENT:WEB:PORT=7449
CLIENT:WEB:SOCKETTIMEOUT=-1
CLIENT:WEB:HOST=localhost
CLIENT:WEB:WATCHDOGTIMEOUT=-1
CLIENT:DRIVER:TYPE=gui_web
CLIENT:GUI:DESKTOPWIDTH=1024
CLIENT:GUI:TASKBAR=true
CLIENT:GUI:DESKTOPHEIGHT=768
```

and the following exception is thrown

```
avax.net.ssl.SSLException: Connection has been shutdown: javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: No trusted certificate found
    at sun.security.ssl.SSLSocketImpl.checkEOF(SSLSocketImpl.java:1529)
    at sun.security.ssl.SSLSocketImpl.checkWrite(SSLSocketImpl.java:1541)
    at sun.security.ssl.AppOutputStream.write(AppOutputStream.java:71)
    at java.io.ObjectOutputStream$BlockDataOutputStream.drain(ObjectOutputStream.java:1877)
    at java.io.ObjectOutputStream$BlockDataOutputStream.setBlockDataMode(ObjectOutputStream.java:1786)
    at java.io.ObjectOutputStream.<init>(ObjectOutputStream.java:247)
    at com.goldencode.p2j.net.NetSocket.<init>(NetSocket.java:79)
    at com.goldencode.p2j.net.SessionManager.createQueue(SessionManager.java:1036)
    at com.goldencode.p2j.net.LeafSessionManager.connectDirect(LeafSessionManager.java:201)
    at com.goldencode.p2j.net.SessionManager.connectDirect(SessionManager.java:587)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:201)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:100)
```

```

    at com.goldencode.p2j.main.ClientDriver.start (ClientDriver.java:201)
    at com.goldencode.p2j.main.CommonDriver.process (CommonDriver.java:398)
    at com.goldencode.p2j.main.ClientDriver.process (ClientDriver.java:95)
    at com.goldencode.p2j.main.ClientDriver.main (ClientDriver.java:267)
Caused by: javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: No trusted certificate found
    at sun.security.ssl.Alerts.getSSLException (Alerts.java:192)
    at sun.security.ssl.SSLSocketImpl.fatal (SSLSocketImpl.java:1937)
    at sun.security.ssl.Handshaker.fatalSE (Handshaker.java:302)
    at sun.security.ssl.Handshaker.fatalSE (Handshaker.java:296)
    at sun.security.ssl.ClientHandshaker.serverCertificate (ClientHandshaker.java:1478)
    at sun.security.ssl.ClientHandshaker.processMessage (ClientHandshaker.java:212)
    at sun.security.ssl.Handshaker.processLoop (Handshaker.java:979)
    at sun.security.ssl.Handshaker.process_record (Handshaker.java:914)
    at sun.security.ssl.SSLSocketImpl.readRecord (SSLSocketImpl.java:1050)
    at sun.security.ssl.SSLSocketImpl.performInitialHandshake (SSLSocketImpl.java:1363)
    at sun.security.ssl.SSLSocketImpl.startHandshake (SSLSocketImpl.java:1391)
    at sun.security.ssl.SSLSocketImpl.getSession (SSLSocketImpl.java:2225)
    at com.goldencode.p2j.net.NetSocket.<init> (NetSocket.java:75)
    ... 9 more
Caused by: sun.security.validator.ValidatorException: No trusted certificate found
    at sun.security.validator.SimpleValidator.buildTrustedChain (SimpleValidator.java:384)
    at sun.security.validator.SimpleValidator.engineValidate (SimpleValidator.java:133)
    at sun.security.validator.Validator.validate (Validator.java:260)
    at sun.security.ssl.X509TrustManagerImpl.validate (X509TrustManagerImpl.java:324)
    at sun.security.ssl.X509TrustManagerImpl.checkTrusted (X509TrustManagerImpl.java:229)
    at sun.security.ssl.X509TrustManagerImpl.checkServerTrusted (X509TrustManagerImpl.java:124)
    at sun.security.ssl.ClientHandshaker.serverCertificate (ClientHandshaker.java:1460)
    ... 17 more

```

X509TrustManager did not validate the server certificate. I found this key security:certificate:validate under clientConfig but didn't find any other usages except for ProcessClientSpawner. I would like to try TLS connection with certificate, but don't know how to manage it for p2j server. The directory.xml provides certificates under <node class="container" name="certificates">.

Constantin, could you look at this, you did "Clean the security caches when client restarts." for SecurityManager, its security cache is used internally and the first primary session with server is done by the programmatic authentication, the second session is used the internal cache data and GuestAccess mode given by

```

<node class="container" name="config">
  <node class="authMode" name="auth-mode">
    <node-attribute name="mode" value="4"/>
    <node-attribute name="retries" value="0"/>
    <node-attribute name="plugin" value="com.goldencode.p2j.security.GuestAccess"/>
  </node>
  <node class="container" name="auth-plugins">
    <node class="container" name="guest_login">
      <node class="string" name="classname">
        <node-attribute name="value" value="com.goldencode.p2j.security.GuestAccess"/>
      </node>
      <node class="string" name="description">
        <node-attribute name="value" value="Non-Interactive Guest Auto-Login"/>
      </node>
      <node class="string" name="option">
        <node-attribute name="value" value="bogus"/>
      </node>
    </node>
  </node>
  <node class="strings" name="resource-plugins">
    <node-attribute name="values" value="com.goldencode.p2j.security.SystemResource"/>
    <node-attribute name="values" value="com.goldencode.p2j.main.EntryPointResource"/>
  </node>
  <node class="string" name="debug-level">
    <node-attribute name="value" value="DATA"/>
  </node>
</node>

```

Maybe it is not the right direction to dig into.

#107 - 11/18/2015 01:00 PM - Sergey Ivanovskiy

I have found a solution to come back to my first investigation #98. The following code fixes the restart issue

```
if (SecurityManager.getInstance() == null && uuid != null)
{
    config.setConfigItem("security", "certificate", "validate", "false");
    config.setConfigItem("security", "trust_mgr", "disable", "true");
}
```

#108 - 11/18/2015 01:02 PM - Sergey Ivanovskiy

It is exactly what we can find if do googling <http://www.rgagnon.com/javadetails/java-fix-certificate-problem-in-HTTPS.html>

#109 - 11/18/2015 02:44 PM - Greg Shah

but there exist the temporary user accounts used to create session by programmatic authentication (com/goldencode/p2j/security/package.html). What is the purpose of this step 7b if the client session can be created directly using another method or can't?

The idea is that we connect through a temporary session (using a **one time** userid and password) in order to pull down the certificate store needed to automatically connect the permanent session.

The temporary accounts can only be used when setup as part of the main operating system login page that comes from the P2J server's jetty (<https://localhost:7443/gui>).

After that one-time use, the temporary account should not be used again. And it should not need to be used again, because all the certificate data is already down at the client.

I may be misremembering things, but I thought we do send the server's cert store to the client so that the client can properly validate the server's certificate. Please review [#1811](#) and look for references to UUID and TemporaryAccount.

In ClientCore SecurityManager clears its cache after restart.

Yes, this is possibly an issue. We need to at least save off the data needed for authentication/reconnect.

```
SERVER:SPAWNER:UUID=0fada8e4-c9a3-496c-8d81-5efe6b97913a
```

This can only be used for the temporary account login. It is effectively the "one-time" use password.

for SecurityManager, its security cache is used internally and the first primary session with server is done by the programmatic authentication, the second session is used the internal cache data and GuestAccess mode given by

The second session is using GuestAccess but in other cases it might not. This isn't really important because I think it would work fine if only we were able to establish the SSL socket without a failure.

I have found a solution to come back to my first investigation #98. The following code fixes the restart issue

I'm nervous about this, because it disables security that I thought we already had (the ability to validate the server connection using the server certificate).

#110 - 11/18/2015 04:36 PM - Sergey Ivanovskiy

I'm nervous about this, because it disables security that I thought we already had (the ability to validate the server connection using the server certificate).

Ok, I will check the cached parameters of the web client security manager. But from the BootstrapConfig provided for the client and from the SecurityManager.needsServerValidation that returns true if the corresponding parameter is present we can conclude that the client security manager doesn't validate server certificates.

```
private boolean needsServerValidation(BootstrapConfig config)
{
    if (config == null)
    {
        config = this.cfg;
    }

    String item = null;

    try
    {
        item = config.getConfigItem("security", "certificate", "validate");
    }

    catch (ConfigurationException cfg)
    {
        // assume no server cert validation
    }

    return "true".equals(item);
}
```

The web client gets only ServerKeyStore keyStore = (ServerKeyStore) spawner.exportData(uuid) for the web server.

#111 - 11/18/2015 05:07 PM - Sergey Ivanovskiy

- File security_issue.txt added

Found that after the client has established the regular session with the server it has ZeroActionTrustManager(X509TrustManager) and the last one accepts any certificates!!! To be precisely the web client security manager has trivial SecurityCache and its TransportSecurity has only ZeroActionTrustManager.

The web client active configuration is below

```
WEB:REFERRER:URL=https://127.0.0.1:7443/gui
SERVER:SPAWNER:UUID=00cbf741-d8b8-4f92-ba92-1b52316ad462
NET:CONNECTION:SECURE=true
SERVER:SECURE_PORT=3334
HOST=localhost
QUEUE:START_THREAD=false
CONVERSATION=true
CLIENT:WEB:PORT=7449
SOCKETTIMEOUT=-1
HOST=localhost
WATCHDOGTIMEOUT=-1
DRIVER:TYPE=gui_web
GUI:DESKTOPWIDTH=1024
TASKBAR=true
DESKTOPHEIGHT=768
```

I think there are 2 cases: it doesn't work as expected or directory.xml is incorrectly configured. Added the diff in order you can check it.

#112 - 11/19/2015 10:14 AM - Sergey Ivanovskiy

Greg, I have no ideas except this one to check if we provide a trust key store for the web client in the bootstrap configuration and the provided trust key store should contain the target server certificate in order it can be validated during the SSL handshake session.

#113 - 11/19/2015 11:49 AM - Greg Shah

Found that after the client has established the regular session with the server it has ZeroActionTrustManager(X509TrustManager) and the last one accepts any certificates!!! To be precisely the web client security manager has trivial SecurityCache and its TransportSecurity has only ZeroActionTrustManager.

This is a consequence of ClientCore.buildTemporaryClientConfig() which sets bc.setConfigItem("security", "trust_mgr", "disable", "true");.

Then the TransportSecurity constructor does this:

```
// for applet code, to disable server certificate validation it will
// take a special trust manager that disables checking since there is
// no local trust store that can be accessed
String txt = bc.getString("security", "trust_mgr", "disable", "false");

if (txt.equalsIgnoreCase("true"))
```

```
{
    tm = new TrustManager[] { new ZeroActionTrustManager() };
}
```

Since `SecurityManager.createInstance()` doesn't reset the `TransportSecurity` by default, the instance with `ZeroActionTrustManager` gets reused.

```
// if on the client and with the same client type config
if (!securityManager.cfg.isServer() && !bc.isServer())
{
    // replace the config with the new one
    securityManager.cfg = bc;

    if (bc.getBoolean("security", "transport", "refresh", false))
    {
        // there are cases when the transport security needs to be invalidated and
        // refreshed. the client config will inform us of such cases.
        securityManager.tranSec = null;
    }
}
```

Our current client truststore approach in the `TransportSecurity` constructor doesn't allow us to provide an in-memory instance of a truststore that we read from the server.

The part I don't understand is how we are successfully creating the "main" session connection (the `connectDirect()` on line 196, after the temporary connection) if we are not also disabling certificate validation? You found above that only the temporary connection disables cert validation. I also don't see anyplace we do that for the main connection. So how does the main connection work that first time?

The web client gets only `ServerKeyStore keyStore = (ServerKeyStore) spawner.exportData(uuid)` for the web server.

I had forgotten that we only use this to download the keystore used to initialize the SSL environment for the client's Jetty embedded web server (see `GenericWebServer.getSslContextFactory()`). This is not used for the main P2J server session connection.

I have no ideas except this one to check if we provide a trust key store for the web client in the bootstrap configuration and the provided trust key store should contain the target server certificate in order it can be validated during the SSL handshake session.

Please answer the question above first. This may indeed be the solution, however I think we will defer this work. For now I want to get a proper understanding of the problem.

Our current client truststore approach in the TransportSecurity constructor doesn't allow us to provide an in-memory instance of a truststore that we read from the server.

I think we can provide truststore by these settings

```
config.setConfigItem("security", "truststore", "bytes", Base64.byteArrayToBase64(keyStore.getKeyStore()));
config.setConfigItem("access", "password", "truststore", keyStore.getKeyStorePassword());
```

because in our case the keyStore provided for the web client contains the "standard" server certificate and the corresponding private key. I tried to use these settings

```
config.setConfigItem("security", "truststore", "bytes", Base64.byteArrayToBase64(keyStore.getKeyStore()));
config.setConfigItem("access", "password", "truststore", keyStore.getKeyStorePassword());
config.setConfigItem("security", "certificate", "validate", "true");
config.setConfigItem("security", "trust_mgr", "disable", "false");
```

but the connection is failed during the validation phase in the SecurityManager. If the idea is correct and the server uses this keys pair for SSL connection, then this issue can be fixed somehow in SecurityManager. Also I rebuilt the server certificates in the directory.xml using this command

```
java -classpath ../../uast/p2j/build/lib/p2j.jar com.goldencode.p2j.security.SSLCertGenUtil ./directory.xml
Initializing service for directory ./directory.xml...
Reading company configuration...
Using 'US' for [C] Country Code.
Using 'Alpharetta' for [L] Locality (city, etc).
Using 'GCD' for [O] Organization.
Using 'Information Technology' for [OU] Organization Unit.
Using 2048 bytes for RSA private key strength.
Using 65337 for RSA public exponent.
Using 'Georgia' for [ST] State or Province Name.
Using 10 years for validity.
Adding account /security/accounts/processes/standard
Adding account /security/accounts/users/bogus
WARNING: alias shared is set for multiple accounts!
Account /security/accounts/users/hs1182snv99on10i ignored - no alias is defined
Account /security/accounts/users/8bmmmlk01t01a30e ignored - no alias is defined
Account /security/accounts/users/g26fi52z5rws3v24 ignored - no alias is defined
Account /security/accounts/users/p7tba8091wz292sm ignored - no alias is defined
Account /security/accounts/users/y1tap1i863579rjs ignored - no alias is defined
Account /security/accounts/users/pl0v361rle99igl4 ignored - no alias is defined
Account /security/accounts/users/0ts4p6e8y307l8pp ignored - no alias is defined
Account /security/accounts/users/leop07a60lx78q1k ignored - no alias is defined
Enter the root CA alias: p2j-root
Generating root CA using [p2j-root] alias...
Generating certificate for account with alias [shared]...
Generating certificate for server with alias [standard]...
Encrypt the in-directory private keys using the same password (yes/no): yes
Private key for root CA [p2j-root] was encrypted using the [redacted] password and the password was NOT SAVED
in the directory. Losing this password will prohibit using the root CA to issue new certificates.
Adding certificate for alias [p2j-root] to the /security/certificates/cas/p2j-root node...
Private key for alias [shared] was encrypted using the [redacted] password and saved in the /security/certific
ates/private-keys/shared node.
Use the access:password:keyentry-shared key to set this password in the server's bootstrap config file and del
ete the /security/certificates/private-keys/shared node from the directory.
Private key for alias [standard] was encrypted using the [redacted] password and saved in the /security/certif
icates/private-keys/standard node.
Use the access:password:keyentry-standard key to set this password in the server's bootstrap config file and d
elete the /security/certificates/private-keys/standard node from the directory.
Adding certificate for alias [shared] to the /security/certificates/peers/shared node...
Adding certificate for alias [standard] to the /security/certificates/peers/standard node...
Save the account private keys in external key-store(s) (yes/no): no
```

```
Save servers' certificates in an external key store (yes/no): no
Save the root CA private key in an external key store (yes/no): no
Done.
WARNING! Any configuration set at the client.xml or server.xml files or via bootstrap config arguments will have priority over the in-directory keys or certificates.
WARNING! The private key encryption passwords for all the account certificates are saved unencrypted in the directory. The root CA's private key can be safely deleted, if it is not required to issue other certificates using this CA.
WARNING! All private keys are encrypted using the same [redacted] password. If needed, delete the 'key-password' nodes from the directory manually, and set this password using the access:password:masterkeyentry bootstrap config at server startup.
```

The part I don't understand is how we are successfully creating the "main" session connection (the connectDirect() on line 196, after the temporary connection) if we are not also disabling certificate validation? You found above that only the temporary connection disables certificate validation. I also don't see anywhere we do that for the main connection. So how does the main connection work that first time?

It works because SecurityManager instance has TransportSecurity object that is reused and the last one is configured with these options

```
config.setConfigItem("security", "certificate", "validate", "false");
config.setConfigItem("security", "trust_mgr", "disable", "true");
```

to accept any certificate. For an example, if you set these settings

```
config.setConfigItem("security", "certificate", "validate", "true");
config.setConfigItem("security", "trust_mgr", "disable", "false");
```

after the first temporary session is created, then the second regular session is failed.

LE: GES redacted keystore passwords from this entry.

#115 - 11/19/2015 12:42 PM - Sergey Ivanovskiy

We can use now this insecure solution and after the trust certificates support will be fixed, we will eliminate it.

#116 - 11/19/2015 02:15 PM - Greg Shah

Agreed. Go ahead with your changes (you can check them in to 2677a).

We will work the rest in [#2860](#).

#117 - 11/19/2015 02:21 PM - Sergey Ivanovskiy

Agreed. Go ahead with your changes (you can check them in to 2677a).

We will work the rest in [#2860](#).

Greg, I have done this schema and it works now and would like to commit within an hour after the code cleanup.

#118 - 11/19/2015 02:40 PM - Greg Shah

OK.

#119 - 11/19/2015 03:23 PM - Sergey Ivanovskiy

- *File certificate_issue_1.txt added*

Greg, please review the committed revision 11065 that added the trust certificate checking.

#120 - 11/19/2015 04:27 PM - Greg Shah

Code Review Task Branch 2677a Revision 11065

The changes are good. I closed [#2860](#) since you have resolved this.

I also checked in some code cleanups in rev 11066. Please take a quick look to see.

I can close this task (2778), right?

#121 - 11/19/2015 04:30 PM - Sergey Ivanovskiy

Yes, it seems that the other issues are fixed for restart.

#122 - 11/19/2015 04:32 PM - Greg Shah

- *Status changed from WIP to Closed*

#123 - 11/16/2016 12:12 PM - Greg Shah

Files

ask-gui.png	27.8 KB	10/26/2015	Sergey Ivanovskiy
displayError_1.txt	914 Bytes	10/27/2015	Sergey Ivanovskiy
invalid_chars.png	23.8 KB	10/27/2015	Sergey Ivanovskiy
restart_1.txt	922 Bytes	10/28/2015	Sergey Ivanovskiy
restart_2.txt	434 Bytes	10/28/2015	Sergey Ivanovskiy
restart_4.txt	8.56 KB	10/30/2015	Sergey Ivanovskiy
workaround_20151102.txt	1.77 KB	11/03/2015	Sergey Ivanovskiy
restart_workaround_20151103.txt	2.83 KB	11/03/2015	Sergey Ivanovskiy
lost_keys_input_fix_20151103_1.txt	7.33 KB	11/03/2015	Sergey Ivanovskiy
lost_keys_input_fix_20151103_2.txt	8.54 KB	11/03/2015	Sergey Ivanovskiy
lost_keys_input_fix_20151103_3.txt	9.06 KB	11/03/2015	Sergey Ivanovskiy
lost_keys_input_fix_20151103_3.txt	9.45 KB	11/03/2015	Sergey Ivanovskiy
lost_keys_input_fix_20151103_4.txt	9.79 KB	11/03/2015	Sergey Ivanovskiy
lost_keys_input_fix_20151104_1.txt	13.6 KB	11/04/2015	Sergey Ivanovskiy
lost_keys_input_fix_20151104_1.txt	11.9 KB	11/04/2015	Sergey Ivanovskiy
11050.txt	613 Bytes	11/13/2015	Sergey Ivanovskiy
realized_1.txt	2.79 KB	11/16/2015	Sergey Ivanovskiy
ask-gui.p_wrong_program_name_in_progress_20151116.png	5.63 KB	11/16/2015	Greg Shah
secure_connection.txt	973 Bytes	11/17/2015	Sergey Ivanovskiy
11056.txt	2.4 KB	11/17/2015	Sergey Ivanovskiy
p2j_runtime_architecture_20141207.pdf	65.4 KB	11/17/2015	Greg Shah
security_issue.txt	1.89 KB	11/18/2015	Sergey Ivanovskiy
sertificate_issue_1.txt	8.72 KB	11/19/2015	Sergey Ivanovskiy