Base Language - Feature #3253

add cross-session publish/subscribe

02/24/2017 10:49 AM - Greg Shah

Status:	WIP	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Constantin Asofiei	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD

Description

Related issues:

Related to Base Language - Feature #1607: implement full named events support	**Closed**	**01/29/2013**	**07/01/2013**
Related to Base Language - Feature #3254: add support for running 4GL on mult...	**WIP**		

History

#1 - 02/24/2017 10:50 AM - Greg Shah

My idea is to provide a generic API (via cross-document messaging) that the embedded JS application can up-call to the converted 4GL and subscribe to an event. When that event fires, it would not call a converted internal procedure, but would call down to the web client and then execute a JS call-back that in invoked on the embedded JS code. The idea is that the embedded JS code only needs to know the event name and we don't need to write any stub 4GL code to make this work.

The idea here is that this is an extension to the ABL facilities, withoutbreaking compatibility.

#2 - 02/24/2017 10:51 AM - Greg Shah

From Constantin:

> I don't think it will be difficult. The tricky part will be for the JS code to respond to data changes from any session, not just current user's session: so, when a PUB will be raised for this event, all JS clients subscribed to it will be notified (this is different than the 4GL approach where PUB/SUB are only for the session).
>
> If the JS responds only for events raised by this user's session, then I don't see much applicability to the approach.

#3 - 02/24/2017 10:52 AM - Greg Shah

I totally agree that providing the cross-session pub/sub is very useful. Please plan to implement both approaches. The subscribe should allow choosing the scope (within-session/cross-session). The within-session approach is "compatible" with the current 4GL implementation. The cross-session approach is an IPC that enables building new application features that the 4GL never supported before. I like it.

Please note that even the within-session approach is useful because it enables multiple parts of the application to act like a single app, even when they are not directly aware of the "connection". This loose coupling is especially useful when we are writing code in JS that needs to integrate with the converted 4GL. Without this, the JS code would have to poll or it would require the user to click a refresh button to make the changes visible. Neither one is an attractive alternative.

#4 - 02/24/2017 10:54 AM - Greg Shah

The updates are being put into 3209e. From Constantin on 07/23/2017:

> Almost done with the JS side, runtime looks good; hope to finish it tomorrow. A short description: semantics of the SUBSCRIBE/PUBLISH when a remote side is using it are the same as the legacy ones; external subscribers are identified as an ExternalResource (a specialized resource which can act as a handle, with no counterpart in the legacy ABL).

#5 - 02/24/2017 10:55 AM - Greg Shah

Question from Constantin:

> I thought that it was possible to just create an AssociativeThread with the proper context and execute the LT.publish API from that thread, when publishing an event to 'global subscribers' (from other clients) - but it doesn't work as I have concurrency issues.
>
> The idea is I need a way to execute this API under a certain client's context; I'm not sure, but I hope posting an ASYNC message to the client's queue will work... if you have any other idea, please let me know.

#6 - 02/24/2017 10:56 AM - Greg Shah

- Related to Feature #1607: implement full named events support added

#7 - 02/24/2017 11:42 AM - Greg Shah

The current named events support is implemented as an in-session dispatching mechanism that directly calls the subscribed internal procedures. This exactly matches the 4GL approach. There is no event queue involved in this solution today. Since everything in the converted 4GL is single-threaded, the event does not need to be queued, the subscribed procedures are just synchronously invoked and the publish does not return until after all subscribers are complete.

Adding cross-session publishing means that we must handle the remote session processing as a notification or a queued event. By nature this is completely asynchronous. The current processing in the remote subscriber can be executing anything, either on the client or the server. Storing the event/notification is needed.

I would like to ensure that one could modify the ABL to use FWD-specific enhanced syntax and be able to enable publishing to remote sessions and subscribing to remote events. Initially we need the support from javascript. Even if we don't implement the ABL side of this right now, I would like the runtime implementation to be enabled for this idea without much (if any) rework.

Your idea to use a client-side queue is one possibility. Was your idea to use the serverEvents queue? I think using the primary (current) queue is too unpredictable due to the asynchronous nature of the event. We don't know where in the processing this will be picked up. I think the server events queue might work. Or we could create a separate queue for this. If we use the server events queue, then integration with the current 4GL code is possible. The processing would happen at the next WAIT-FOR or PROCESS-EVENTS, like any other event processing.

It is not clear how the JS request/notification would integrate with this idea. Please share more of your thoughts on this side of the solution so I can understand it better.

I will respond on the AssociatedThread part of the question next.

#8 - 02/24/2017 12:16 PM - Constantin Asofiei

Greg Shah wrote:

> It is not clear how the JS request/notification would integrate with this idea. Please share more of your thoughts on this side of the solution so I can understand it better.

I have not introduced yet PUBLISH support from the JS, but is not hard to add the APIs - do you want me to do so?

Current approach is this:

1. JS side calls SUBSCRIBE - if global, it registers into a global registry, where a list of subscriber data is kept for each session
2. 4GL side calls PUBLISH:
   - if the subscription is from the same client, then it calls the JS side immediately.
   - Otherwise, it uses TC.invokeLater with a task which calls the JS side (for PUBLISH notification) next time event processing is performed. Note that serverEvents does not match this usecase, because when a ServerEvent is processed, it goes back into the server to execute a piece of code. Looking at the code, PROCESS-EVENTS will not pick up the EventManager.serverEvents, as TC.processEvents uses EventManager.queue.

#9 - 02/24/2017 12:39 PM - Greg Shah

I assume you are trying to use the AssociatedThread in the subscribing client's context. You can't create such a thread at the time of the publish, so I assume you would have to create it at the time of the subscribe.

In regard to the concurrency problems, I can certainly see this as a likely scenario. The key issue here is that the state belonging to the ABL compatibility portions of the session (e.g. TransactionManager, BlockManager, the persistence layer/Hibernate...) are all designed with the concept that they will only ever be running on a single thread, so they are not thread-safe. In our use of these for server events, socket events and so forth, I think we keep the processing scoped so narrowly as to avoid concurrently conflicts with the ABL state. In addition, we leverage the ASYNCH message processing support in the Dispatcher to safely send an event to the client asynchronously.

Thinking more about it, I'm not sure how safe the current approach is for the scheduler user cases that we have. They all run under the server's context and if more than one were to initialize the legacy environment simultaneously, then I think it would quickly lead to a catastrophic result.

> Otherwise, it uses TC.invokeLater with a task which calls the JS side (for PUBLISH notification) next time event processing is performed.

I think that if you create an AssociatedThread at the time of SUBSCRIBE and that thread simply calls some thread-safe API (e.g. in NamedEventManager) that only returns when there is a match to the event, then that code could do the "local" portion of the PUBLISH as you are planning. You will have to keep the processing to a minimum. So the thread should probably be limited to getting the notification to the client side/setting up the invokeLater() and then iterate its loop. I think using the registerAynchThread()/deregisterAynchThread() will be key for this.

That NamedEventManager API would also need to return when the session dies and the thread's loop will need to cleanly exit.

My biggest concern with this is that it is specific to the JS use case.

> Note that serverEvents does not match this usecase, because when a ServerEvent is processed, it goes back into the server to execute a piece of code.

Won't it (or something like it) be needed for the case where we enable enhanced ABL code to SUBSCRIBE to a global event?

> Looking at the code, PROCESS-EVENTS will not pick up the EventManager.serverEvents, as TC.processEvents uses EventManager.queue.

I think this is probably a latent bug for some use cases. If I understand correctly, use cases like read-response and procedure-complete are supposed to be handled there.

> I have not introduced yet PUBLISH support from the JS, but is not hard to add the APIs - do you want me to do so?

Yes, I think so.

#10 - 02/24/2017 01:08 PM - Greg Shah

- Related to Feature #3254: add support for running 4GL on multiple threads in a single session added

#11 - 02/28/2017 05:57 AM - Constantin Asofiei

Greg/Hynek: please review 3209e rev 11218. It modifies PUBLISH/SUBSCRIBE/UNSUBSCRIBE so that:

1. these can be used from external applications (which embed a P2J client).
2. clients can register for named events globally, so they are notified by any context which raises this event.

When global subscriptions are used by PUBLISH, tasks are posted to a server thread which will execute the task in the subscriber's context (by starting an AssociatedThread with that client's context, which will send an async request to the client). In this case, the PUBLISH is not i.e. executed immediately, but:

- for legacy subscription, posted via a ServerEvent - this is not finished, just commented out
- for external subscriptions, posted via an OS event

This way, the client will pick up the event and execute it next time it will process the events.

Eric: Hotel GUI rev 92 contains changes to show how the PUBLISH/SUBSCRIBE/UNSUBSCRIBE can be used from JS.

Next step is to do full ETF regression testing.

#12 - 02/28/2017 07:20 AM - Hynek Cihlar

Constantin Asofiei wrote:

> Greg/Hynek: please review 3209e rev 11218.

Pretty good, I didn't any issues except I would consider using a Java Executor service for dispatching the publish tasks in GlobalEventSubscriptions. As of now the number of threads is virtually unlimited and with improper server resource allocation and unexpected load could lead to a halt.

Regarding the invokeLater() TODO in TC.publish(). I believe invokeLater() could be used if you ensure proper dispatching of the event queue.

#13 - 02/28/2017 11:27 AM - Constantin Asofiei

Hynek Cihlar wrote:

> Constantin Asofiei wrote:
>
> > Greg/Hynek: please review 3209e rev 11218.
>
> Pretty good, I didn't any issues except I would consider using a Java Executor service for dispatching the publish tasks in GlobalEventSubscriptions. As of now the number of threads is virtually unlimited and with improper server resource allocation and unexpected load could lead to a halt.

Hm... Executor I don't think can be used, because the threads will not have the proper P2J context. An alternative would be to not execute the task in a different thread, but directly in the GlobalEventSubscriptions thread - the tasks do not take much time, they just call an API on the client-side to post an event.

> Regarding the invokeLater() TODO in TC.publish(). I believe invokeLater() could be used if you ensure proper dispatching of the event queue.

I don't understand what you mean here... I didn't dig deeper, but from what I saw the event was posted, but not picked up.

#14 - 02/28/2017 11:34 AM - Hynek Cihlar

Constantin Asofiei wrote:

> Hynek Cihlar wrote:

> Constantin Asofiei wrote:
>
> > Greg/Hynek: please review 3209e rev 11218.
>
> Pretty good, I didn't any issues except I would consider using a Java Executor service for dispatching the publish tasks in GlobalEventSubscriptions. As of now the number of threads is virtually unlimited and with improper server resource allocation and unexpected load could lead to a halt.

Hm... Executor I don't think can be used, because the threads will not have the proper P2J context. An alternative would be to not execute the task in a different thread, but directly in the GlobalEventSubscriptions thread - the tasks do not take much time, they just call an API on the client-side to post an event.

You can give the Executor a thread factory and there instantiate and setup the threads, see java.util.concurrent.ThreadFactory.

> Regarding the invokeLater() TODO in TC.publish(). I believe invokeLater() could be used if you ensure proper dispatching of the event queue.

I don't understand what you mean here... I didn't dig deeper, but from what I saw the event was posted, but not picked up.

Yes, it wasn't picked up because there was no event bracket to dispatch the event.

#15 - 02/28/2017 12:56 PM - Constantin Asofiei

Hynek Cihlar wrote:

> You can give the Executor a thread factory and there instantiate and setup the threads, see java.util.concurrent.ThreadFactory.

Even so, we can't ensure which thread ends up executing which client. Another appraoch would be for each client to have its own dedicated thread to process these events - this way we have a fix number of threads.

> Yes, it wasn't picked up because there was no event bracket to dispatch the event.

We can't process this in an event bracket immediately, but in the next event processing from a i.e. WAIT-FOR or PROCESS-EVENTS.

#16 - 02/28/2017 12:57 PM - Greg Shah

Code Review Task Branch 3209e Revision 11218

Overall, it seems pretty good. I really like how you made the p2j.remote.js and p2j.embedded.js code generic (array based instead of a if/else if/else).

1. This implementation only supports a String[] for parameters. This is much more limited than the ABL PUB/SUB feature set. We will probably need to expand this to match the full syntax at some point. We can wait until the rest of the legacy support goes in.

2. Do you intend for ProcedureManager.GlobalEventSubscriptions.listen() to eat all InterruptedException instances? What happens when the server needs to exit? I think it will continue to try to process things, which might yield cases of race conditions.

I think the logic is OK when a spurious wakeup occurs, although this is dependent upon the classes in use (they have to deal with the empty iterator without raising a problem).

#17 - 02/28/2017 01:05 PM - Constantin Asofiei

Greg Shah wrote:

> 1. This implementation only supports a String[] for parameters. This is much more limited than the ABL PUB/SUB feature set. We will probably need to expand this to match the full syntax at some point. We can wait until the rest of the legacy support goes in.

The String is used for serialization from JS side into the P2J runtime (and vice-versa). When calling the legacy procedure, the argument will be automatically converted to the proper type, if the type doesn't match. On the JS side, it's a little more difficult, as each argument will have to be converted explicitly to the expected type.

> 2. Do you intend for ProcedureManager.GlobalEventSubscriptions.listen() to eat all InterruptedException instances? What happens when the server needs to exit? I think it will continue to try to process things, which might yield cases of race conditions.

This thread is a daemon thread, so when JVM shuts down, it will not stop JVM to shut down. But I'll think of a way for this thread to die gracefully.

#18 - 03/01/2017 02:16 PM - Eric Faulhaber

Greg Shah wrote:

> I have not introduced yet PUBLISH support from the JS, but is not hard to add the APIs - do you want me to do so?

Yes, I think so.

Was this implemented?

I have a use case for JS to publish an event in the embedded hotel sample, in order to update the state of the embedded P2J client. The JS side seems to post the event, but I can't seem to catch it on the P2J side.

The publish from P2J to JS is working nicely! I'm using a global subscription, though I've only tested it locally so far.

#19 - 03/01/2017 03:36 PM - Constantin Asofiei

Eric Faulhaber wrote:

> I have a use case for JS to publish an event in the embedded hotel sample, in order to update the state of the embedded P2J client. The JS side seems to post the event, but I can't seem to catch it on the P2J side.

Yes, it works. How did you use the SUBSCRIBE in the 4GL code? This should be something like subscribe to "loggedin" anywhere., with a procedure like this defined to catch the event:

```
procedure loggedin.
   def input param p1 as char.
   def input param p2 as char.

   message p1 p2 view-as alert-box.
end.
```

and the JS side can call this to post it: p2j.embedded.publish("loggedin", loggedinResourceEvent, ["hotel2", "hotel2"]);, but this can be done only after the 4GL side has subscribed.

On 4GL side, you can't really use the external resource ID without some tricks to get it from the JS side, so SUBSCRIBE is simpler to use ANYWHERE option, and not specify a certain PUBLISHER-HANDLE. If you want to specify it, you need to invoke a dedicated 4GL procedure which gets as argument the external resource ID and call SUBSCRIBE with that.

#20 - 03/01/2017 04:00 PM - Eric Faulhaber

I have this in the "Main Block" of avail-rooms-frame.w:

```
SUBSCRIBE TO "capacityZoom" ANYWHERE.
```

This converts to:

```
NamedEventManager.subscribeAnywhere(thisProcedure(), "capacityZoom");
```

I have set a breakpoint on this and it is getting called.

I've added this internal procedure to avail-rooms-frame.w:

```
&ANALYZE-SUSPEND _UIB-CODE-BLOCK _PROCEDURE capacityZoom fFrameWin 
PROCEDURE capacityZoom :
/*------------------------------------------------------------------------------
  Purpose:     Set start and end dates in UI and check room availability.
  Parameters:  zoomStart
               Start date of capacity zoom date range. Corresponds with start
               date (check in) for a room availability check.
               zoomEnd
               End date of capacity zoom date range. Corresponds with the day
               before end date (check out) for a room availability check.
  Notes:       This procedure is called from a subscription to the "capacityZoom"
               event. Subscription event parameters are sent as strings, not
               dates; they must be converted to dates before updating the UI and
               performing the availability check.
------------------------------------------------------------------------------*/

DEF INPUT PARAMETER zoomStart AS CHAR.
DEF INPUT PARAMETER zoomEnd AS CHAR.

  RUN check-availability (INPUT DATE(zoomStart), INPUT (DATE(zoomEnd) + 1)).

END PROCEDURE.

/* _UIB-CODE-BLOCK-END */
&ANALYZE-RESUME
```

I've added this to the JS code:

```
var capacityZoomResourceEvent;
...
me.dateRangeSelected = function dateRangeSelected(start, end)
{
   var a = ehotel.formattedDate(start);
   var b = ehotel.formattedDate(end);
   ...
   // notify subscribers that a date range on the hotel capacity heatmap has been selected
   p2j.embedded.publish("capacityZoom", capacityZoomResourceEvent, [a, b]);
}
```

a and b are date strings formatted to match Progress' format string "99/99/9999".

I've debugged the JS code up to the point where p2j.embedded.js posts the message, but a breakpoint in the converted capacityZoom procedure never gets hit.

Do I need to initialize capacityZoomResourceEvent somehow?

#21 - 03/01/2017 04:45 PM - Constantin Asofiei

Eric Faulhaber wrote:

> I've debugged the JS code up to the point where p2j.embedded.js posts the message, but a breakpoint in the converted capacityZoom procedure never gets hit.
>
> Do I need to initialize capacityZoomResourceEvent somehow?

Hmm... I assume this is undefined in your case. Look how loggedinResourceEvent was used - this is currently initialized via a p2j.embedded.subscribe's callback. You need the same for capacityZoomResourceEvent.

I think I'll add an API to build an external resource on 4GL side... as this might be useful beyond PUB/SUB.

#22 - 03/01/2017 10:07 PM - Eric Faulhaber

- File publish_js_to_p2j_param_fix_20170301a.patch added

Constantin Asofiei wrote:

> Eric Faulhaber wrote:
>
> > I've debugged the JS code up to the point where p2j.embedded.js posts the message, but a breakpoint in the converted capacityZoom procedure never gets hit.
> >
> > Do I need to initialize capacityZoomResourceEvent somehow?
>
> Hmm... I assume this is undefined in your case. Look how loggedinResourceEvent was used - this is currently initialized via a p2j.embedded.subscribe's callback. You need the same for capacityZoomResourceEvent.
>
> I think I'll add an API to build an external resource on 4GL side... as this might be useful beyond PUB/SUB.

This throws ArrayStoreException for the publish event:

```
Caused by: java.lang.ArrayStoreException
        at java.lang.System.arraycopy(Native Method)
        at com.goldencode.p2j.util.ControlFlowOps$InternalEntryCaller.valid(ControlFlowOps.java:5909)
        at com.goldencode.p2j.util.ControlFlowOps.validArguments(ControlFlowOps.java:4752)
        at com.goldencode.p2j.util.ControlFlowOps.invokeImpl(ControlFlowOps.java:4338)
        at com.goldencode.p2j.util.ControlFlowOps.invoke(ControlFlowOps.java:3403)
        at com.goldencode.p2j.util.ControlFlowOps.invokeImpl(ControlFlowOps.java:4051)
        at com.goldencode.p2j.util.ControlFlowOps.invokeImpl(ControlFlowOps.java:3980)
        at com.goldencode.p2j.util.ControlFlowOps.invokeInImpl(ControlFlowOps.java:3936)
        at com.goldencode.p2j.util.ControlFlowOps.invokeInWithMode(ControlFlowOps.java:877)
        at com.goldencode.p2j.util.ProcedureManager$LegacySubscription.publish(ProcedureManager.java:3531)
        at com.goldencode.p2j.util.ProcedureManager.publish(ProcedureManager.java:957)
        at com.goldencode.p2j.util.ProcedureManager.publishExternal(ProcedureManager.java:1277)
        at com.goldencode.p2j.util.NamedEventManager.publishExternal(NamedEventManager.java:584)
        at com.goldencode.p2j.ui.LogicalTerminal.publishExternal(LogicalTerminal.java:14632)
        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:709)
        at com.goldencode.p2j.net.Conversation.block(Conversation.java:364)
        at com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:300)
        at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1128)
        at com.goldencode.p2j.net.Queue.transact(Queue.java:599)
        at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:223)
        at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:163)
        at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1425)
        at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:97)
        at com.sun.proxy.$Proxy17.waitFor(Unknown Source)
        at com.goldencode.p2j.ui.LogicalTerminal.waitFor(LogicalTerminal.java:6280)
        at com.goldencode.p2j.ui.LogicalTerminal.waitFor(LogicalTerminal.java:6050)
        at com.goldencode.hotel.Emain.lambda$execute$109(Emain.java:1013)
        at com.goldencode.hotel.Emain$$Lambda$44/84847686.body(Unknown Source)
        ...
```

At ControlFlowOps:5909, we try to use System.arraycopy to copy from an Object array containing character elements into a String array. This is due to the use of ControlFlowOps$ArrayArgumentResolver, which accepts a String[] object via the c'tor parameter of type Object..., then returns it during argument resolution. This appears isolated to this new code path, where the argument array comes in as a String array over the GUI web socket.

The attached patch fixes the problem. I didn't want to commit it to 3209e, since that branch already has passed ETF testing.

#23 - 03/02/2017 08:05 AM - Constantin Asofiei

Eric, see 3209e rev 11220 - it contains fixes for:

1. the PUBLISH issue in your previous note
2. added p2j.embedded.createResource API, to create a resource without relying on the SUBSCRIBE callback; the functionality behind SUBSCRIBE remains the same (i.e. the resource gets built if is not valid)
3. some other fixes

As I noted, ETF testing passed, but MAJIC runtime testing fails early; CTRL-C just blocked.

Greg: should I look into it now?

#24 - 03/02/2017 08:10 AM - Constantin Asofiei

Constantin Asofiei wrote:

> Eric, see 3209e rev 11220 - it contains fixes for:

PS: all the changes in 11220 do not affect ETF, as they are in JS code or in APIs not in use by it.

#25 - 03/02/2017 08:32 AM - Greg Shah

> Greg: should I look into it now?

Yes. We need to get 3209e into the trunk this week.

#26 - 03/13/2017 12:29 PM - Eric Faulhaber

Constantin Asofiei wrote:

> added p2j.embedded.createResource API, to create a resource without relying on the SUBSCRIBE callback; the functionality behind SUBSCRIBE remains the same (i.e. the resource gets built if is not valid)

What is the idiom to use this API?

I see the createResource function in p2j.embedded.js creates and sends a msg object. Do I just store that msg object as the resource event in the callback function I provide to createResource, or do I have to do something more specific with it?

#27 - 03/13/2017 12:41 PM - Constantin Asofiei

Eric Faulhaber wrote:

> Constantin Asofiei wrote:
>
> > added p2j.embedded.createResource API, to create a resource without relying on the SUBSCRIBE callback; the functionality behind SUBSCRIBE remains the same (i.e. the resource gets built if is not valid)
>
> What is the idiom to use this API?

I've added it so that you are not required to use the subscribe API to create an external resource.

> I see the createResource function in p2j.embedded.js creates and sends a msg object. Do I just store that msg object as the resource event in the callback function I provide to createResource, or do I have to do something more specific with it?

You don't need to use it in your current implementation, but the usage is similar to the subscribe callback - save the resource ID in some var or do other work with it. For example, the callback function provided to createResource can do something like this:

```
   p2j.embedded.createResource(createCallback);
   function createCallback(msg)
   {
      var payload = msg["payload"];
      var someResourceId = msg["payload"]["resource-id"];

      // save the resource-id in some var or initialize other work, i.e. create SUBSCRIBE for this resource, o
r anything else which might be added in the future
   };
```

#28 - 09/03/2019 10:20 AM - Constantin Asofiei

I completely forgot to add cross-session publish/subscribe from between FWD sessions - global subscribe works only from JS currently. I'm adding it now to 4124a, it won't take much.

#29 - 09/03/2019 11:52 AM - Constantin Asofiei

- Status changed from New to WIP

- Assignee set to Constantin Asofiei

This was added in 4124a rev 11506. This one can be closed now.

#30 - 09/03/2019 12:38 PM - Greg Shah

> There is no change in the syntax of the PUBLISH statement - this will automatically notify all subscribers, local or in other clients.

This is a security issue. Code that does not intend to publish across sessions, now does so without any notice. There is an inherent issue of leaking data (both the event occurence and any parameters). I think the publish code needs to be only global if updated with the GLOBAL keyword. It does require some changes to the 4GL, but it is safer.

> This argument validation is done at the publisher's side - any mismatch in the above is a programming error and will abend the FWD client.

What is the exception thrown? Is it something like IllegalArgumentException? I'd prefer to raise STOP.

#31 - 09/03/2019 12:53 PM - Constantin Asofiei

Greg Shah wrote:

> This is a security issue. Code that does not intend to publish across sessions, now does so without any notice.

OK, I'll add it.

> > This argument validation is done at the publisher's side - any mismatch in the above is a programming error and will abend the FWD client.
>
> What is the exception thrown? Is it something like IllegalArgumentException?

Yes.

> I'd prefer to raise STOP.

Will do.

#32 - 09/03/2019 01:42 PM - Constantin Asofiei

Review fixes done in 4124a rev 11507; wiki is updated, too.

#33 - 09/03/2019 02:10 PM - Constantin Asofiei

We will need to update Hotel GUI's publish statement (where it sends the notification for the JavaScript-subscribed event), to add the GLOBAL option. Otherwise that will no longer work. This needs to be after 4124a is in trunk.

Files

publish_js_to_p2j_param_fix_20170301a.patch	2.91 KB	03/02/2017	Eric Faulhaber